UNIVERSITY OF ALBERTA

REAL-TIME VEHICLE ROUTING AND SCHEDULING IN DYNAMIC AND STOCHASTIC TRAFFIC NETWORKS

by

Liping Fu ©

A thesis submitted to the Faculty of Graduate Studies and Research in partial fulfillment of the requirements for the degree of Doctor of Philosophy

in

Transportation Engineering

Department of Civil and Environmental Engineering

Edmonton, Alberta
Fall 1996

National Library
of Canada

Acquisitions and
Bibliographic Services Branch

395 Wellington Street
Ottawa, Ontario
K1A 0N4

Bibliothèque nationale
du Canada

Direction des acquisitions et
des services bibliographiques

395, rue Wellington
Ottawa (Ontario)
K1A 0N4

Your file Votre référence

Our file Notre référence

The author has granted an irrevocable non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of his/her thesis by any means and in any form or format, making this thesis available to interested persons.

L'auteur a accordé une licence irrévocable et non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de sa thèse de quelque manière et sous quelque forme que ce soit pour mettre des exemplaires de cette thèse à la disposition des personnes intéressées.

The author retains ownership of the copyright in his/her thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without his/her permission.

L'auteur conserve la propriété du droit d'auteur qui protège sa thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

ISBN 0-612-18038-7

Canada

UNIVERSITY OF ALBERTA

Library Release Form

NAME OF AUTHOR: **LIPING FU**

TITLE OF THESIS: **Real-time Vehicle Routing and Scheduling in Dynamic and Stochastic Traffic Networks**

DEGREE: **Doctor of Philosophy**

YEAR THIS DEGREE GRANTED: **1996**

Permission is hereby granted to the University of Alberta Library to reproduce single copies of this thesis and to lend or sell such copies for private, scholarly, or scientific research purposes only.

The author reserves all other publication and other rights in association with the copyright in the thesis, and except as hereinbefore provided, neither the thesis nor any substantial portion thereof may be printed or otherwise reproduced in any material form whatever without the author's prior written permission.

Liping Fu

611E Michener Park
Edmonton T6H 5A1
Alberta, Canada

Date: July 3, 1996

UNIVERSITY OF ALBERTA

FACULTY OF GRADUATE STUDIES AND RESEARCH

The undersigned certify that they have read, and recommend to the Faculty of Graduate Studies and Research for acceptance, a thesis entitled "*Real-time Vehicle Routing and Scheduling In Dynamic and Stochastic Traffic Networks*" by **Liping Fu** in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Transportation Engineering

Dr. L. R. Rilett

Dr. S. Teply

Professor A. E. Peterson

Dr. S. M. AbouRizk

Dr. M. J. Zuo

Dr. M. Van Aerde

Date: June 7, 1996

To

my wife, Shirley Ying Guo, for the “optimal” schedules she has given me every day

&

my daughters, Connie and Dannie, for the “dynamic and stochastic” fun they have created.

ACKNOWLEDGEMENT

There are many individuals and organizations whose valuable assistance contributed to the completion of this thesis, to them I gratefully acknowledges. I particularly wish to recognize the following contributions:

- My supervisor, Dr. Laurence R. Rilett for his valuable guidance, assistance and support over the last three years.
- My co-supervisor, Dr. Stan Teply for his valuable assistance and advice.
- The members of my Ph. D. Committee, Professor A. E. Peterson, Dr. S. M. AbouRizk, Dr. M. J. Zuo, and my external reader, Dr. M. Van Aerde for their guidance and constructive criticisms in the final stages of this research.
- Thanks to Darryl Mullen , Ida Tse and Raymond Dai, who took the time to read the thesis and provided many useful comments.
- Thanks to the Transportation Department of City of Edmonton, who was always cooperative in providing information and materials.
- Financial support provided by the University of Alberta and the Natural Science and Engineering Research Council of Canada (NSERC) is recognized and appreciated.
- Special thanks to my friend Ian Baker and his wife Kathleen Baker for their friendship, generosity and support which helped me get through a tough time during my first year in Canada.
- My parents, Shiwen and Zhuqing Fu, my brother, Weidong, my sister, Weiqing, and my parents-in-law, Gongyi and Yuying Guo in P. R. China, who have supported and inspired me during my stay in Canada.
- Finally, it would not be possible to complete this thesis without the continuous understanding and encouragement provided by my wife.

ABSTRACT

Central to both in-vehicle route guidance systems (RGS) and automated vehicle dispatching systems (AVDS) is the vehicle routing and scheduling component which is required to find “optimal” routes and schedules in real-time for individual RGS-equipped vehicles and AVDS fleet vehicles in urban traffic networks. This thesis is motivated by the realization of the potential importance of explicitly considering the dynamic and stochastic nature of travel times within the vehicle routing and scheduling procedures in both RGS and AVDS and the need to develop efficient routing algorithms that can operate successfully in real time..

The dynamic and stochastic nature of the link travel time under three typical traffic conditions in urban traffic networks are first investigated through various theoretical and statistical procedures. The shortest path problem (SPP) with dynamic and stochastic link travel time and the dial-a-ride problem (DARP) with dynamic and stochastic O-D travel time are formulated and their respective solution algorithms are developed. These models and algorithms are then used to analyze the influences of the uncertainties of the travel times on the routing and scheduling results. The techniques from the artificial intelligent field (AI), including heuristic search strategies and artificial neural networks (ANN), are applied in vehicle routing and travel time estimation procedures to improve the computational efficiency of the routing and scheduling algorithms for real-time operation purposes.

The theoretical and computational analyses indicates that the consideration of the dynamic and stochastic nature of travel times in the SPP will result in different "optimal" paths as compared to a deterministic model. It is found that the dynamic and stochastic nature of travel times has a significant effect on the routing and scheduling results of the DARP. The heuristic routing and scheduling algorithms developed in this thesis are extensively tested and evaluated using an actual network from the City of Edmonton, Alberta and the results indicate that these algorithms are applicable in real-time operation systems such as RGS and AVDS.

CONTENTS

CHAPTER 1 INTRODUCTION 1

1.0 OVERVIEW 1

1.1 IN-VEHICLE ROUTE GUIDANCE SYSTEM (RGS) AND REAL TIME AUTOMATED VEHICLE DISPATCHING SYSTEM (AVDS): OVERVIEW AND PROBLEMS 3

1.1.1 In-vehicle Route Guidance System (RGS) 3

1.1.2 Automated Vehicle Dispatching System (AVDS) 8

1.2 STUDY OBJECTIVES 13

1.3 RESEARCH FRAMEWORK AND METHODOLOGY 14

1.3.1 Investigate the Link Travel Time Distribution Patterns under Various Traffic Conditions 15

1.3.2 Formulate and Solve the SPP 16

1.3.3 Formulate and Solve the DARP 17

1.3.4 Develop Heuristic Vehicle Routing Algorithms 18

1.3.5 Testing 19

1.4 ANTICIPATED CONTRIBUTIONS OF THE RESEARCH 19

1.5 ORGANIZATION OF THE THESIS 20

REFERENCES 22

CHAPTER 2 LITERATURE REVIEW 24

2.0 INTRODUCTION 24

2.1 ESTIMATION AND PREDICTION OF LINK TRAVEL TIME 25

2.1.1 ADVANCE System 26

2.1.2 ALI_SCOUT System 28

2.1.3 Pathfinder and TravTek Systems 31

2.2 SHORTEST PATH PROBLEM AND ALGORITHMS 32

2.2.1 Shortest Path in a Static and Deterministic Network 33

2.2.1.1 Label Correcting Algorithms: 35
2.2.1.2 Label Setting Algorithms: 36
2.2.1.3 Computation Studies on Label Correcting and Label Setting Algorithms: 37
2.2.2 Shortest Path in a Dynamic Network 38
2.2.3 Shortest Path in a Stochastic Network 40
2.2.4 Shortest Path in a Dynamic and Stochastic Network 40
2.3 K SHORTEST PATH PROBLEMS AND ALGORITHMS 43
2.4 DIAL-A-RIDE PROBLEMS AND SOLUTION METHODS 45
2.4.1 Subscriber DIAL-A-RIDE Problem and Algorithms 47
2.4.2 Real-time DIAL-A-RIDE Problem and Algorithms 50
2.5 SUMMARY 52
REFERENCES 54

CHAPTER 3 DYNAMIC AND STOCHASTIC LINK TRAVEL TIME 59

3.0 INTRODUCTION 59
3.1 LINK TRAVEL TIME PATTERN UNDER UNINTERRUPTED UNDERSATURATED FLOW CONDITIONS 61
3.1.1 Running Time Distribution: Some Empirical Evidence 62
3.1.2 Theoretical Derivation of the Running Time Distribution 64
3.2 LINK TRAVEL TIME DISTRIBUTION UNDER A TRAFFIC SIGNAL CONTROL CONDITION 68
3.2.1 A Simulation Model 70
3.2.2 Verification of the Simulation Model 73
3.2.3 Intersection Delay Distribution: A Sensitivity Analysis 76
3.2.3.1 Vehicle delay distribution vs. traffic volume 76
3.2.3.2 Vehicle delay distribution vs. green time 77
3.2.3.3 Vehicle delay distribution vs. the quality of the progression 78
3.3 LINK TRAVEL TIME DISTRIBUTION UNDER INCIDENT CONGESTION 83
3.3.1 Assumptions and Annotation 84
3.3.2 Probability Distribution of Incident Delay 89
3.3.3 Mean and Variance of Incident Delay 92
3.3.4 Expected Incident Delay: A Comparison to the Deterministic Incident Delay Model 93
3.3.5 Variation of Incident Delay 96
3.3.6 Incident duration: prior and posterior probability distribution 99

3.4 CONCLUSIONS 101

REFERENCES 103

CHAPTER 4 ESTIMATION OF ROUTE TRAVEL TIME IN DYNAMIC AND STOCHASTIC TRAFFIC NETWORKS 105

4.0 INTRODUCTION. 105

4.1 PROBLEM DEFINITION 107

4.2 THE MEAN AND VARIANCE OF ROUTE TRAVEL TIME 109

- 4.2.1 Mean of Route Travel Time 111
- 4.2.2 Variance of Route Travel Time 115

4.3 LINK TRAVEL TIME SMOOTHING 119

4.4 SOLUTION QUALITY OF THE APPROXIMATION MODELS: A SIMULATION STUDY 123

- 4.4.1 Simulation Procedure 124
- 4.4.2 The Edmonton Network 126
- 4.4.3 Simulation Scenarios 126
- 4.4.4 Approximation Quality and Sensitivity Analysis 130
  - 4.4.4.1 General performance 130
  - 4.4.4.2 Sensitivity to the link travel time COV 131
  - 4.4.4.3 Sensitivity to the dynamic pattern of the link travel time 132
  - 4.4.4.4 Sensitivity to the PDF of the link travel time 133

4.5 CONCLUSIONS 139

REFERENCES 141

CHAPTER 5 ESTIMATION OF EXPECTED MINIMUM PATHS IN DYNAMIC AND STOCHASTIC TRAFFIC NETWORKS 142

5.0 INTRODUCTION 142

5.1 DYNAMIC AND STOCHATIC SHORTEST PATH PROBLEM: DEFINITION AND PROPERTIES 144

5.2 A HEURISTIC ALGORITHM TO CALCULATE THE EXPECTED SHORTEST PATH 149

5.3 COMPUTATIONAL ANALYSIS 151

5.4 CONCLUSIONS 155

REFERENCES 156

CHAPTER 6 HEURISTIC SHORTEST PATH ALGORITHMS 158

6.1 INTRODUCTION 158

6.2 THE SHORTEST PATH PROBLEM AND OPTIMAL ALGORITHMS 160

6.2.1 Label Correcting Algorithm 162

6.2.2 Label Setting Algorithm 163

6.2.3 Computational Performance of the Optimal Shortest Path Algorithms 164

6.3 HEURISTIC SHORTEST PATH SEARCH METHODS 165

6.3.1 Limiting the Search area 166

6.3.1.1 Branch Pruning Method 166

6.3.1.2 A* Algorithm 172

6.3.2 Decomposing the Search Problem 176

6.3.2.1 Bi-directional Search Method 176

6.3.2.2 Subgoal Method 179

6.3.3 Limiting the Search Links 181

6.3.3.1 Hierarchical Search Method 182

6.4 COMPUTATIONAL STUDY 185

6.4.1 Performance of Branch Pruning Algorithms 188

6.4.2 Performance of A* Type Algorithms 190

6.4.3 Performance of Bi-directional Search Algorithms 191

6.4.4 Selection of Heuristic Algorithms 192

6.5 CONCLUSIONS 198

REFERENCES 201

CHAPTER 7 ESTIMATION OF DYNAMIC AND STOCHASTIC O-D TRAVEL TIME USING ARTIFICIAL NEURAL NETWORKS (ANN) 203

7.0 INTRODUCTION 203

7.1 NEURAL NETWORK BASED TRAVEL TIME ESTIMATION MODEL 206

7.1.1 ANN Network Topology 206

7.1.2 Data Representation 208

7.1.3 Training and Testing Examples 209

7.1.4 Training, Testing and Results 213
7.2 COMPARISON OF TRAVEL TIME ESTIMATION USING AN ANN AND A REGRESSION MODEL 220
7.2.1 Data 220
7.2.2 Models 220
7.2.3 Results 221
7.3 COMPARISON OF COMPUTATIONAL EFFICIENCY BETWEEN THE ANN METHOD AND SHORTEST PATH ALGORITHMS 221
7.4 CONCLUSIONS 224
REFERENCES 225

CHAPTER 8 DIAL-A-RIDE VEHICLE ROUTING AND SCHEDULING WITH DYNAMIC AND STOCHASTIC O-D TRAVEL TIME . 226

8.0 OVERVIEW 226
8.1 DIAL-A-RIDE VEHICLE ROUTING AND SCHEDULING PROBLEM: MODELS 228
8.1.1 Objectives Related to Service Operator 230
8.1.2 Objectives Related to Each Service Vehicle and Driver 231
8.1.3 Objectives Related to Customers 231
8.1.3.1 Satisfaction from the pick-up/drop-off time 231
8.1.3.2 Satisfaction from the ride time 237
8.1.4 Problem Formulation 239
8.2 HEURISTIC DIAL-A-RIDE VEHICLE ROUTING AND SCHEDULING ALGORITHM 243
8.2.1 Feasibility Test 244
8.2.2 Optimization 245
8.2.3 O-D Travel Time Estimation Methods 245
8.3 COMPUTATIONAL ANALYSIS 246
8.3.1 Test Problems 246
8.3.2 System Performance Vs. the Objective Function Parameter a_2 248
8.3.3 System Performance Vs. Objective Function Parameter a_3 251
8.3.4 Computational Efficiency Vs. O-D Travel Time Estimation Methods 253
8.4 CONCLUSIONS 256
REFERENCES 257

CHAPTER 9 CONCLUSIONS AND RECOMMENDATIONS 259

9.0 INTRODUCTION 259

9.1 CONCLUSIONS 259

9.1.1 Dynamic and Stochastic Link Travel Time 260

9.1.2 Route Travel Time in Dynamic and Stochastic Networks 262

9.1.3 Shortest Path Problem in Dynamic and Stochastic Networks 263

9.1.4 Heuristic Shortest Path Algorithms 264

9.1.5 Dynamic and Stochastic OD Travel Time Estimation Using Artificial Neural Networks 266

9.1.6 Dial-A-Ride Vehicle Routing and Scheduling with Dynamic and Stochastic OD Travel Time 268

9.2 RECOMMENDED FURTHER RESEARCHS 269

9.2.1 On Dynamic and Stochastic Link Travel Time 269

9.2.2 On Shortest Path Problem in Dynamic and Stochastic Networks 270

9.2.3 On Heuristic Shortest Path Algorithms 271

9.2.4 On Dynamic and Stochastic OD Travel Time Estimation Using Artificial Neural Networks (ANN) 271

9.2.5 On Dial-A-Ride Vehicle Routing and Scheduling with Dynamic and Stochastic OD Travel Time 272

APPENDIX A COMPUTATIONAL EFFICIENCY OF THE BRANCH PRUNNING ALGORITHM: AN EXPLANATORY MODEL.... 273

APPENDIX B COMPUTATIONAL EFFICIENCY OF THE HIEARCHICAL SEARCHING ALGORITHM: AN EXPLANATORY MODEL. 278

APPENDIX C ARTIFICIAL NEURAL NETWORKS: AN INTRODUCTION. 284

GLOSSARY 288

LIST OF FIGURES

Page

Figure 1-1 A conceptualization of a in-vehicle route guidance system (RGS) 5

Figure 1-2 A conceptualization of a real-time automated vehicle dispatching system (AVDS) 10

Figure 1-3 A combined framework 15

Figure 3-1 The running time distribution: surveyed vs. mathematical 63

Figure 3-2 The running time distributions as compared to normal distribution 67

Figure 3-3 Relationship between the time horizon, time interval and cycle length 70

Figure 3-4 Cyclic arrival pattern at the approach 71

Figure 3-5 The intersection delay distributions: simulated vs. the surveyed 75

Figure 3-6 The mixed PMF and PDF of the vehicle delay under different traffic volumes 80

Figure 3-7 The relationship between the mean and standard deviation of the vehicle delay and traffic volume 80

Figure 3-8 Vehicle delay distribution as compared to normal distribution when v/c=0.95 81

Figure 3-9 The mixed PMF and PDF of the vehicle delay under different green time 81

Figure 3-10 Relationship between the mean and standard deviation of the delay and green time ... 82

Figure 3-11 The mixed PMF and PDF of the vehicle delay under different quality of progression ... 82

Figure 3-12 Relationship between the mean and standard deviation of the delay and the platoon ratio ... 83

Figure 3-13 Queuing model of the incident delay and the related parameters ... 87

Figure 3-14 The density function of the incident duration and the parameters related to incident delay ... 87

Figure 3-15 The mixed PMF and PDF of the incident delay ... 92

Figure 3-16 Estimation of the expected incident delay: deterministic model vs. stochastic model ... 95

Figure 3-17 Estimation error of the incident delay by a deterministic model ... 97

Figure 3-18 Estimation of the incident delay: mean and standard deviation ... 98

Figure 3-19 Prior and posterior distribution function of the incident duration ... 101

Figure 4-1 A route from origin node s to destination node g including link (i, i+1) .. 108

Figure 4-2 The effect of the link travel time pattern on the estimation of the expected link travel time ... 114

Figure 4-3 The effect of the link travel time pattern on the estimation of the link travel time variance ... 118

Figure 4-4 Schematic illustration of Link travel time smoothing models ... 122

Figure 4-5 Illustration of the relation between the link travel time smoothing scheme with the arrival time pattern and link travel time interval 123

Figure 4.6 The Edmonton Network 127

Figure 4.7 Time variation patterns of the link travel time 128

Figure 4.8 Probability distribution patterns of the link travel time 129

Figure 4.9 A Comparison of the first order model with the second order model 134

Figure 4.10 Estimation of route travel time standard deviation: naive approximation model vs. simulation results 134

Figure 4-11 Estimation of route travel time standard deviation: the second order approximation model vs. the simulation results 135

Figure 4-12 Estimation of route travel time standard deviation: the first order model vs. the second order model 135

Figure 4-13 Relationship between the route travel time mean estimation quality with link travel time covariance 136

Figure 4-14 Relationship between the route travel time standard deviation estimation error with link travel time covariance 136

Figure 4-15 Relationship between the route travel time mean estimation error with link travel time pattern 137

Figure 4-16 Relationship between the estimation error of the route travel time standard deviation with link travel time pattern 137

Figure 4-17 Relationship between the route travel time mean estimation quality with link travel time distribution 138

Figure 4-18 Relationship between the route travel time variance estimation quality with link travel time distribution ... 138

Figure 5-1 A path from origin node s to destination node g including link (i, j) in a traffic network ... 145

Figure 5-2 A simple dynamic and stochastic network ... 147

Figure 5-3 An acyclic network ... 148

Figure 5-4 Link travel time pattern ... 152

Figure 5-5 Solution quality vs. K value ... 153

Figure 5-6 Solution quality vs. K value ... 154

Figure 5-7 CPU time vs. K value ... 154

Figure 6-1 A schematically illustration of the pruning power of the branch pruning algorithms ... 169

Figure 6-2 A schematically illustration of the pruning power of the A* algorithms ... 175

Figure 6-3 A schematically illustration of the pruning power of the bi-directional search algorithms ... 177

Figure 6-4 Pruning power of using a sub-goal ... 181

Figure 6-5 A schematically illustration of the shortest path search procedure in a two level hierarchical network ... 184

Figure 6-6 Computational efficiency of the branch pruning algorithms ... 193

Figure 6-7 Solution quality of the branch pruning algorithms ... 194

Figure 6-8 CPU time vs. OD travel time: branch pruning algorithms ... 194

Figure 6-9 Computational efficiency of the A* type algorithms 195

Figure 6-10 Solution quality of the A* type algorithms 195

Figure 6-11 CPU time vs. OD travel time: A* type algorithms 196

Figure 6-12 CPU time vs. parameter m: bi-directional algorithms 196

Figure 6-13 Relative error vs. parameter m: bi-directional algorithms 197

Figure 6-14 Relative performance of the heuristic algorithms 198

Figure 7-1 ANN topology for O-D travel time estimation 207

Figure 7-2 The Edmonton network and subareas 211

Figure 7-3 Dynamic link travel time pattern 212

Figure 7-4 Covariance of O-D travel time by a separate network model: actual value vs. estimated value as a function of trip length 217

Figure 7-5 Covariance of O-D travel time by a joined network model: actual value vs. estimated value as a function of trip length 217

Figure 7-6 Learning progress curve for an AM net 218

Figure 7-7 Actual mean travel time vs. travel time predicted by the AM net 218

Figure 7-8 Actual travel time pattern vs. estimated travel time pattern: two OD pairs 219

Figure 7-9 The relationship between the computation time and the travel time: A* shortest path algorithm 223

Figure 8-1 Time window with random arrival time 234

Figure 8-2 Customer's ride time condition 238

Figure 8-3 The relationship between the number of vehicles required and the parameter a2 ... 249

Figure 8-4 The relationship between the expected vehicle productivity and the parameter a_2 ... 250

Figure 8-5 The relationship between the expected average time deviation and the parameter a_2 ... 250

Figure 8-6 The relationship between the number of vehicles required and the parameter a_3 ... 252

Figure 8-7 The relationship between the expected vehicle productivity and the parameter a_3 ... 252

Figure 8-8 The relationship between the expected average excess ride time and the parameter a_3 ... 253

Figure 8-9 Relationship between the computational time and problem size: subscriber DARP ... 255

Figure 8-10 Relationship between the computational time and problem size: real-time DARP ... 255

Figure A-1 A schematically illustration of the search area of the LS algorithm and BP_LS algorithm in idealized network ... 274

Figure A-2 Computational efficiency of the branch pruning algorithm ... 277

Figure B-1 A hierarchical Euclidean network ... 279

Figure B-2 The computational efficiency of the hierarchical label setting algorithm .. 282

Figure C-1 A typical ANN processing element (PE) ... 285

LIST OF TABLES

Page

Table 3-1 Simulation parameters for model verification 74

Table 6-1 Heuristic algorithms and their acronyms 186

Table 7-1 Link travel time covariance in different subareas 212

Table 7-2 Training data examples (original data) 213

Table 7-3 Comparison of prediction errors of the ANN model and the regression model 222

Table 7-4 Computation time of the shortest path algorithms and the ANN model 223

LIST OF ABBREVIATIONS

AI:	Artificial Intelligence
ANN:	Artificial Neural Network
AVDS:	Automated Vehicle Dispatching System
BP:	Branch Pruning Algorithm
COV:	Coefficient of Variation
DARP:	Dial-A-Ride Problem
ITS:	Intelligent Transportation System
LC:	Label Correcting Algorithm
LS:	Label Setting Algorithm
O-D:	An Origin Location to a Destination Location
PDF:	Probability Density Function
PMF:	Probability Mass Function
RGS:	Route Guidance System
SPP:	Shortest Path Problem
TIC:	Traffic Information Center

CHAPTER 1

INTRODUCTION

1.0 OVERVIEW

In the past 30 years there has been significant interest in applying advanced technologies to solve the problems associated with the surface transportation system — the field now commonly referred to as Intelligent Transportation Systems (ITS). It is generally thought that by gathering, processing, displaying and communicating information in a real-time fashion, advances in the surveillance, telecommunication and computer technologies provide an opportunity to improve the transportation system by reducing congestion, pollution and accidents. One of the major ideas behind the ITS is to help transportation network users maximize their travel related satisfactions based on real-time traffic information.

There are two general categories of the ITS field that are directly oriented to transportation network users. The first category is R[illegible] Guidance System (RGS), which focuses on guiding individual travelers or fleet vehicles to their specified destination by providing intelligent and safe advice on which transportation mode they should use, when is the best time for them to depart, and which routes or roads they should take.

The second category is Commercial Vehicle Operations (CVO) and Advanced Public Transportation Systems (APTS), in which one major component is the real-time Automated Vehicle Dispatching System (AVDS) for commercial vehicle and public transportation fleet management. The objective of the AVDS is to help the dispatcher better manage fleet operations by providing drivers with better routes and schedules based on the real-time information about traffic status, demands and vehicle locations. The end result is an improvement in the reliability and efficiency of carrier pick-up/delivery operations.

While they could have very different structures in system architecture, both RGS and AVDS typically have common components. In particular, a vehicle route optimizer is required to solve the underlying vehicle routing problems in a real-time fashion. In an RGS, the routing problem is to find the "optimal" routes in an urban traffic network for users who want to travel from one location (origin) to another one (destination). Conversely, the vehicle routing problem within an AVDS consists of determining the "optimal" pickup and/or delivery routes and schedule for vehicles required to visit a number of spatially and temporally dispersed locations in an given area.

To model and solve these routing problems, one of the most important pieces of information needed is the anticipated link travel time. In an RGS, link travel times are directly used to find the "optimal" route from an origin to a destination. In an AVDS, the link travel times are used to calculate the anticipated travel time from one location (origin) to another location (destination), or O-D travel time. The O-D travel times are then used to find the optimal routes and schedules for the fleet vehicles.

Due to inherent fluctuation of travel demands, unpredictable occurrences of traffic incidents and changes in weather conditions, the travel times in an urban traffic environment may be significantly dynamic and stochastic.

This research is motivated by the realization of the potential importance of explicitly considering the dynamic and stochastic nature of travel times within the vehicle routing and scheduling procedures in both RGS and AVDS and the need to develop efficient routing algorithms that can operate successfully in real time.

1.1 IN-VEHICLE ROUTE GUIDANCE SYSTEM (RGS) AND REAL-TIME AUTOMATED VEHICLE DISPATCHING SYSTEM (AVDS): OVERVIEW AND PROBLEMS

1.1.1 In-vehicle Route Guidance System (RGS)

The in-vehicle RGS are intended to collect real-time information on the status of the traffic network system and communicate the information on the "optimal" routes to RGS-equipped vehicles. The broad strategic goals of the RGS are to improve the network efficiency and reduce traffic congestion, accidents and environment pollution. However, the specific objective of most RGS under development is to allow the equipped vehicles to save travel time when they travel from one location (origin) to another (destination).

Attracted by the potential benefits from the in-vehicle route guidance systems (King and Mast, 1987; U.S.DOT, 1990; Rilett, 1992), various countries are currently experimenting with real-time in-vehicle route guidance systems. In the USA, the major projects in progress include the Travtek program in Orlando, Florida (Rillings and Lewis, 1991), the ADVANCE program in Chicago, Illinois (Boyce, 1991) and the Pathfinder program in Los Angles, California (Mammano, 1991). The Europeans are working on the ALISCOUT systems in Great Britain and Germany (Catlink, 1989), while Japan is experimenting with the Vehicle Information and Control System (VICS) (Takaba, 1991).

Although these systems vary in complexity, they can be illustrated by a conceptualization shown in Figure 1.1. As shown in Figure 1.1, the traffic information used in an RGS is usually managed by a traffic information center (TIC). The TIC receives data from two sources: 1) historical traffic data that contain average travel times by time of day, day of week, season of year, different weather conditions, etc., and 2) real-time traffic data that are collected from the network. The real-time data can be obtained from variety of sources including traffic detectors embedded throughout the network, and probe vehicles which are able to transmit travel time data to the TIC. There are also some other data sources such as police reports of incidents. The real-time data from different data sources can then be combined to generate current link travel time estimates in the network. With the historical and real-time traffic data as inputs, a prediction model is then used to estimate the travel time on each link in the traffic network for a future time horizon, for example, one hour ahead. When a route guidance task is required, the "route optimizer" will try to find the “optimal” routes for the user based on certain objective(s).

Generally there are two types of routing objectives: user optimal and system optimal. In the "user optimal" sense, the “optimal” routes are calculated by considering only the individual RGS user’s objectives such as minimizing travel time, minimizing travel

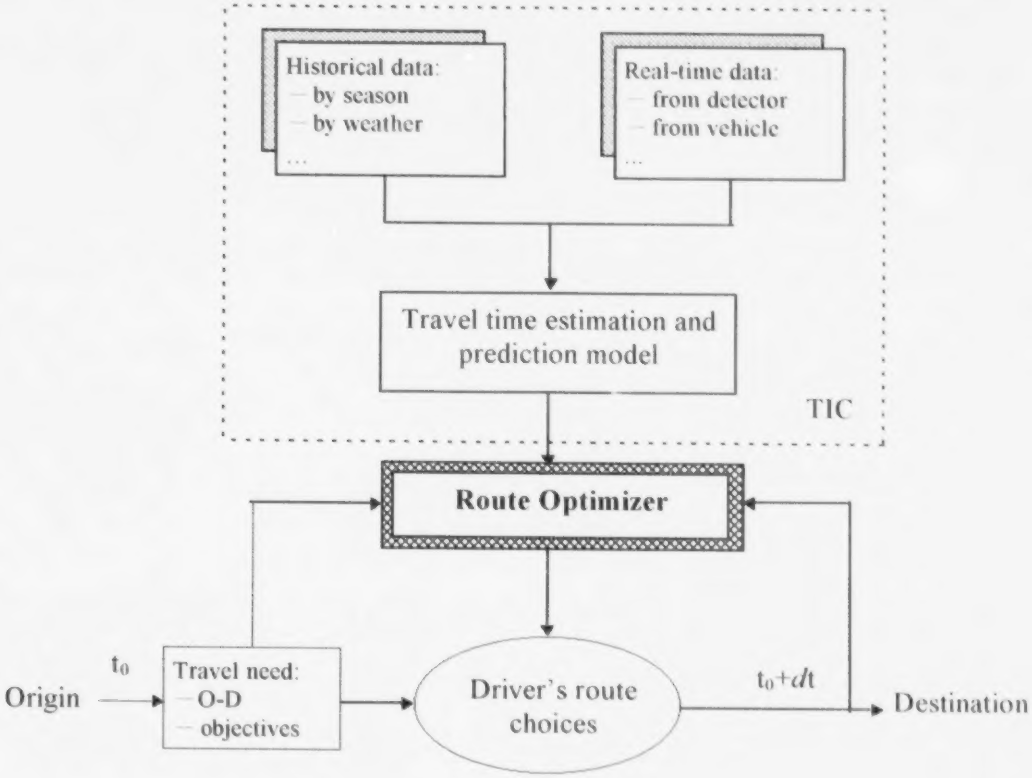

Historical data:
— by season
— by weather
...
Real-time data:
— from detector
— from vehicle
...
Travel time estimation and prediction model
TIC
Route Optimizer
t_0
Origin
Travel need:
— O-D
— objectives
Driver's route choices
t_0+dt
Destination

Figure 1-1 A conceptualization of a in-vehicle route guidance system (RGS)

cost and maximizing safety. In "system optimal" systems, routes are chosen to maximize the benefit of system wide users including unguided drivers, community as well as guided drivers. Although the latter routing objective would be more preferred by the RGS operating authorities, the majority of experimental RGS programs currently under development use the "user optimal" objective because it is more likely to be accepted by the individual users. The RGS with the former routing objective will be the focus of this research and can be specifically defined as finding the path which is anticipated to have the minimum travel time from an origin (for example, current vehicle location) to a destination in a traffic network. This problem is generally referred to as a "shortest path problem" (SPP).

It should be noted in the conceptualization of an RGS as presented above that in order to solve the SPP, one of the most important pieces of information needed is the anticipated link travel times in the underlying traffic network. The treatment of the link travel time also influences how the SPP would be modeled. For example, if the link travel times are assumed to be constant values, then the SPP is basically a static and deterministic problem and the optimal path can be found using traditional shortest path algorithms (Rilett, 1992).

In an urban traffic environment, the link travel times may be highly dynamic and uncertain because of the inherent fluctuation of travel demands, unpredictable occurrences of traffic incidents and changes in weather conditions. The uncertainties in link travel times is problematic in that, theoretically, there is no means of identifying the minimum travel time path among all the paths between two locations until after the trips are made.

Consequently, the SPP itself needs to be redefined and the applicability of the traditional shortest path algorithms needs to be re-examined. For most RGS under development, these problems are circumvented by implicitly assuming that the average link travel may be used in place of the random variables that represent travel time. The shortest paths found in this type of network are used as the expected and hence optimal shortest paths. This method, although simple, has two potential problems. First, there is uncertainty regarding the solution quality of the identified path compared to the optimal solution. That is, there could be a difference between the expected shortest path directly found in a network with dynamic and uncertain link travel times and the "expected" shortest path in the same network but the uncertain link travel times are replaced by average link travel times (Hall, 1986). Secondly, it does not consider the individual driver's attitude toward risk under uncertainty. For example, minimizing the expected travel time may no be the sole objective. It is very plausible that the variation of the travel time and the probability of lateness at the destination may also be critical in determining a driver's route choice. For example, a driver may choose a route that has a 5 minute longer travel time if the variance associated with that trip is small in order to be sure he/she arrives at the destination on time. It is therefore necessary to study the potential cost and benefit for a RGS to incorporate both dynamic and uncertain natures of link travel times in the route optimization procedure. Specifically, the following problems need to be addressed:

i) How should the dynamic and stochastic attributes of link travel times be represented? How can the link travel time estimates be calculated with real-time information?

ii) How can the SPP be modelled when the link travel times are dynamic and stochastic? Specially, how should the "optimal" route be defined?

iii) How would the uncertainty of the link travel time influence the route choice? When is it prudent to ignore the stochastic attributes by using the average link travel times in routing procedure? What kind of link travel time information should be used to calculate the "optimal" path?

iv) Would the SPP be too complicated to be solved optimally in real-time after explicitly modelling the dynamic and uncertain nature of the link travel times? Would real-time operational solution methods exist for the more realistically modeled SPP?

1.1.2 Real-time Automated Vehicle Dispatching System (AVDS)

The real-time AVDS, as part of advanced fleet management system, focuses on using real-time information on traffic condition and current vehicle locations in the vehicle routing and scheduling optimization process. The primary objective is to improve the efficiency and reliability of the fleet operation by dispatching and scheduling the vehicles in a real-time fashion. It has been realized that real-time data based vehicle routing and scheduling algorithms will provide more efficient routes with less total vehicle travel time. Therefore, fleet vehicles can be more efficiently used and customer service can be more accurately controlled. In addition, the service operator will provide better service by responding more quickly to changes in system status such as a new service request, vehicle breakdowns and unexpected traffic incidents (Stone *et al.*, 1993).

Based on fleet operation characteristic, AVDS can be classified into two categories. One is developed in single stop distribution operations, in which each vehicle is only assigned to serve one demand once at a time. Examples include taxi systems, emergency vehicle services, etc. In this type of operation system, the fundamental function of the "route optimizer" is to solve many shortest path problems and therefore the related problem is similar to the ones in RGS.

Another category occurs in multiple stop distribution operations, where each vehicle may be assigned to serve several demands. There are numerous applications that have this type of service operation. Typical examples include school bus systems, shared-ride dial-a-ride paratransit systems, mail delivery systems, snow plough vehicles, etc. Among them, the shared-ride dial-a-ride paratransit system (or demand responsive transit system) is becoming the first application area of the AVDS in North America (Stone *et al.*, 1993) and will be used as an example application in this research.

Although a number of automated paratransit systems and software have been developed during the past decade, there is not a single system which fully uses real-time travel time data in their vehicle routing and scheduling process. With the undergoing development of the technologies associated with the ITS, it is anticipated that real-time traffic data will be available as part of the major resources provided by traffic information management systems. This thesis will assume that an AVDS would use real-time travel time data as part of the inputs to the vehicle routing and scheduling optimization process; Therefore, the focus will be on how to apply these data to improve the vehicle scheduling and thus system service reliability.

A conceptualization of an AVDS is shown in Figure 1.2. In this hypothetical AVDS, most components are fully automated. At the beginning of the service operation, routes and schedules are prepared and given to each driver to serve the customers who request service in advance (for example, one day in advance). During the operation, a customer calls on the telephone to request a pick-up/drop-off trip, and as soon as the reservation clerk has entered the customer's ID or name, his/her service is verified by the computer. Within a few seconds the computer determines which vehicle to assign the trip to, calculates the new vehicle schedule and informs the reservation clerk when the vehicle will provide the service to the customer. The trip assignment and schedule calculation are conducted by the "route optimizer" based on the information of current vehicle locations from the vehicle locating system (VLS), anticipated link travel times from the traffic

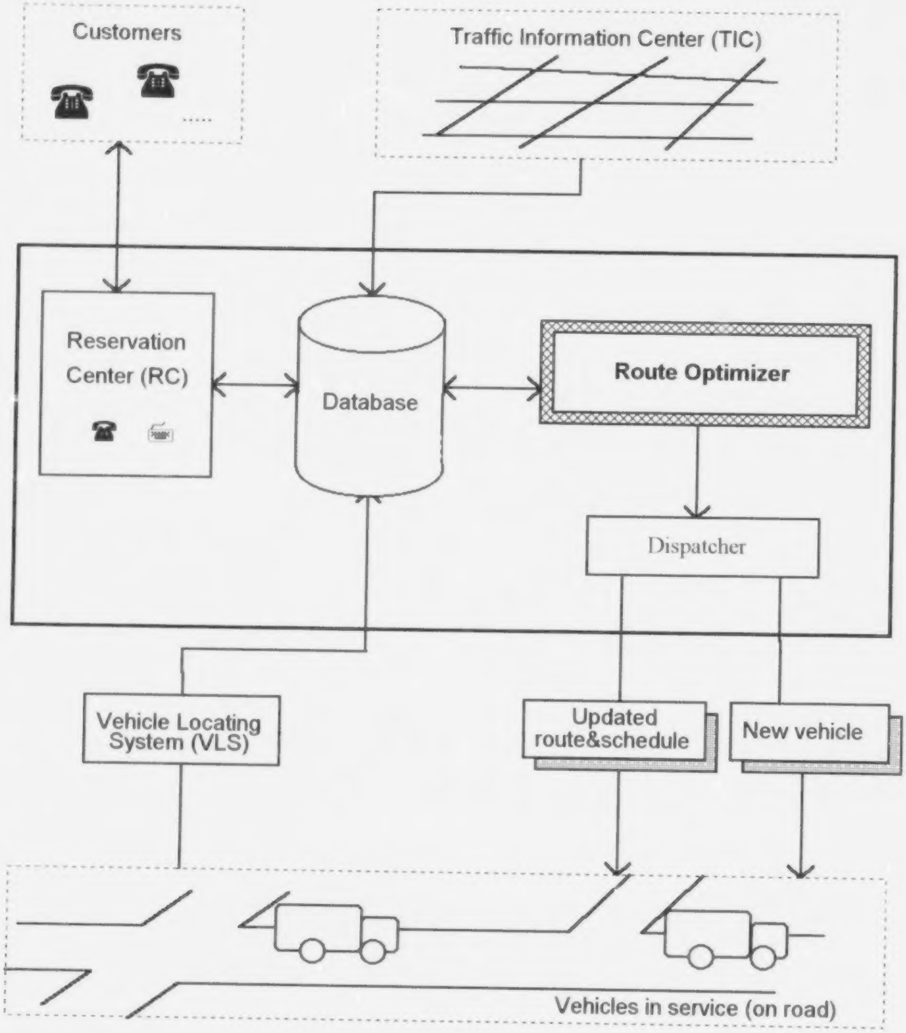

Customers
.....
Traffic Information Center (TIC)
Reservation Center (RC)
Database
Route Optimizer
Dispatcher
Vehicle Locating System (VLS)
Updated route&schedule
New vehicle
Vehicles in service (on road)

Figure 1-2 A conceptualization of a real-time automated vehicle dispatching system (AVDS)

information center (TIC) and customers' information. For each vehicle in service, the travel direction or detailed travel path from a given location to another is also given by the operation center or simply by a RGS.

In this type of operation, the central problem concerns the method used to determine the optimal pickup and drop-off routes and times for a fleet of vehicles, which are required to carry customers between specified origins and destinations. This problem belongs to the well-known "vehicle routing and scheduling problem" and has been specifically referred to as the "dial-a-ride problem" (DARP).

As seen in the operation process shown above, there are indeed two types of DARP involved. One is called advance request DARP (or subscriber DARP) which mainly arises at the beginning of the operation. The vehicle routing and scheduling objective is to assign all the booked customers to the empty vehicles. Because the customers are known in advance, the "route optimizer" can have plenary time (for example overnight) to develop the routes and schedules. Therefore, the computational efficiency of the solution algorithm for this situation is not a critical issue.

The second problem is called demand responsive DARP (or real-time DARP) which arises when a new customer calls for immediate service. At the time of request, the vehicles are following their prearranged routes and schedules to pick-up or deliver customers. At this time, some earlier customers may have been delivered to their destination and, hence, need not be considered in this problem. The remaining earlier customers who have each been assigned to a vehicle are either waiting for pick-up or are on board to their destinations. The problem is to determine the insertion of the new customer into the previous routes and schedules and the new routes and schedules after the customer is inserted. In order to preserve the stability of the routes and schedules, it is usually required that the previous trip assignment and visiting sequence for each vehicle should be kept unchanged as much as possible. For example, the change in route and schedule can be restricted only to the vehicle that the new customer is assigned to. Although this problem is relatively easy to solve compared to the advance request DARP, efficient solution algorithms are still required to make the solution procedure operational in real-time.

While the associated technologies are readily available for an AVDS, a major requirement for successful implementation is to model and solve the real-time DARP. The DARP has been historically modeled in a static and deterministic manner in the sense that the travel time from an origin location to a destination location (or O-D travel time) is assumed to be constant. As described in last section, the O-D travel times in an urban traffic network are inherently dynamic and uncertain. Ignoring the dynamic and stochastic nature of the travel times may result in sub-optimal solutions to the DARP, or more seriously, solutions violating customers' time windows. Therefore, the most realistic model would be one that integrates both the dynamic and stochastic attributes of the O-D travel times into the vehicle routing and scheduling procedure. The following problems need to be addressed:

i) How should the dynamic and stochastic attributes of the O-D travel times be represented? What method should be used to estimate and predict the dynamic and uncertain O-D travel times (or parameters) based on the real-time traffic information in an accurate and quick manner?

ii) How can the DARP be modeled more realistically when the O-D travel times are dynamic and uncertain? Essentially how should the system operator's routing objective and customers' service time requirement be defined?

iii) How would the uncertainty of the O-D travel time influence the route choice? When is it acceptable to ignore the stochastic component of O-D travel time by using an average value in routing procedure?

iv) How can the DARP be solved effectively and efficiently after it explicitly considers the dynamic and uncertain nature of the O-D travel times? Would real-time operational solution methods exist for the more realistically modeled DARP?

1.2 STUDY OBJECTIVES

The initial objective of this research is to evaluate the potential benefits and cost of incorporating the dynamics and uncertainty of the link travel time into the vehicle routing models which arise from two ITS application areas: in-vehicle route guidance system (RGS) and automated vehicle dispatching system (AVDS).

The research will first focus on modeling the link travel time distribution patterns in an urban traffic network under both recurring congestion and incident congestion situations. The uncertainty of the link travel times will be assumed to be caused by random factors and therefore the link travel times will be modeled as random variables

Solution algorithms and methods will then be developed to calculate the "optimal" routes and O-D travel times in a network with dynamic and stochastic link travel times. Artificial intelligence techniques will be applied in the solution method to improve their computation efficiency. Finally, the research will focus on formulating and solving the DARP with dynamic and stochastic OD travel time model. Real-time operational solution methods to this problem will be developed.

With the models and algorithms developed, the research will try to address the following questions:

i) What type of route optimization framework should be used to model the SPP and DARP when the travel times are dynamic and stochastic?

ii) How would the uncertainties of the link travel times affect the route choices in RGS and AVDS? Could they be ignored or is it sufficient to use the average travel times in the routing procedure?

iii) If the stochastic influence is significant, are there any solution methods which will be efficient enough to be implemented in real-time operations?

1.3 RESEARCH FRAMEWORK AND METHODOLOGY

This research will systematically investigate the two real-time vehicle routing problems, the SPP and DARP, in a dynamic and stochastic environment. Figure 1-3 schematically illustrates the combined research framework and the relationship between the underlying problems. The research will first investigate the dynamic link travel time

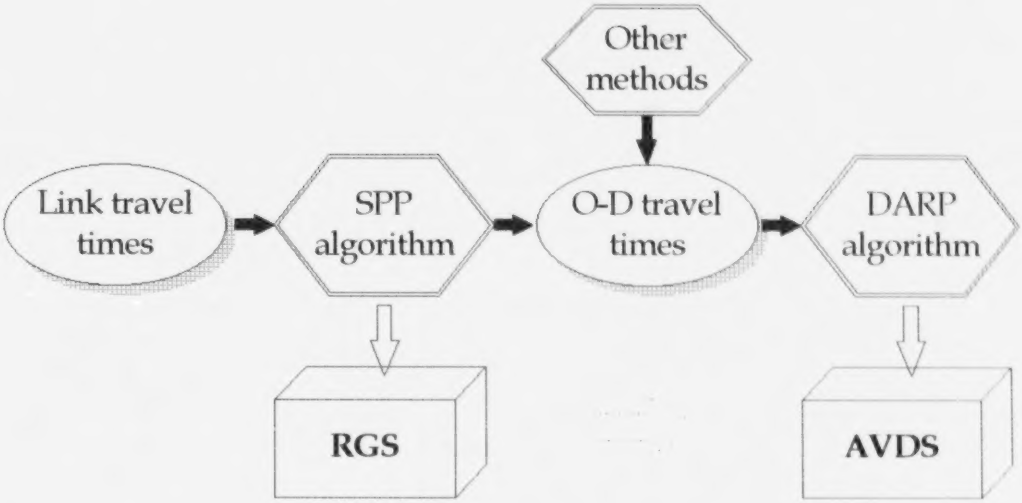


Figure 1-3 A combined framework

distribution patterns. Next, both the SPP with different link travel time models and the DARP under different OD travel time models will be formulated. Solution methods to these problems will be developed and results from different models will be compared in terms of both solution quality and computational effort. For real-time application purposes, this research will also explore the feasibility of applying certain Artificial Intelligence techniques (AI) in the vehicle routing procedures. The detailed discussions are presented in the following sections.

1.3.1 Investigate the Link Travel Time Distribution Patterns under Various Traffic Conditions

Each link is assumed to operate under two traffic conditions: the regular traffic condition and the incident congestion condition. This research will first examine the probability distribution patterns of the link travel time under regular traffic condition using field data from the City of Edmonton and a simulation model. The link travel time distributions will be modeled using mathematical distributions and associated parameters. A basic tenet of this work is that the mathematical distribution used should be as simple as possible for two primary reasons. First, the stochastic behavior of link travel times in a traffic network is a function of many variables including link type, traffic control, etc. Consequently, it is an extremely complicated procedure to estimate the appropriate distribution for a given situation. Secondly, the use of a complicated distribution means using more parameters and therefore more information needs to be stored or communicated for real-time applications.

Under incident congestion traffic conditions, the travel time on a link may have two components: running time on the link and the waiting time in the queue (queuing delay). It is expected that the distribution of link travel time under traffic incident congestion can be established based on historical and real-time information on traffic volume, incident duration and capacity reduction.

1.3.2 Formulate and Solve the Shortest Path Problem (SPP)

The stochastic nature of the SPP ensures that they may be defined in many ways. For the objectives specified in this research, the following two models will be considered:

SPP-1: Find the expected shortest path when the dynamic random travel time is replaced by the dynamic expected travel time;

SPP-2: Directly find the expected shortest path in a network with dynamic and stochastic link travel time;

It will be shown that the first model can be solved by using a dynamic shortest path algorithm. For the second model, its computational tractability will be examined first and subsequently a heuristic algorithm will be developed to solve this problem.

1.3.3 Formulate and Solve the Dial-A-Ride Problem (DARP)

In order to model the DARP in a shared-ride dial-a-ride transit system, the following operational setting will be used:

i) There are a fixed number of vehicles available;

ii) The operation objective is to minimize a combined disutility of the operator and customers. The operator's disutility is defined as a linear function of total vehicle travel time. The disutility of customers is defined as a quadratic function of customer's extra ride time because of other customers and deviations from desired pick-up/delivery times;

iii) A specific level of service must be maintained such that (a) the customer's ride times will not exceed a pre-specified maximum and (b) the time of pick-up and delivery of customers will not deviate from their desired pick-up or delivery time by more that pre-specified amounts. With stochastic travel time, these two conditions will be modified to reflect the associated uncertainty;

With above operation settings, the research will focus on the following two demand responsive DARP models:

DARP-1: Replace the dynamic random travel time with the dynamic expected travel time and find the routes and schedules for the vehicles which will minimize the combined disutility of the operator and customers under specified service constraints;

DARP-2: Directly find the routes and schedules for the vehicles which will minimize the expected combined disutility of the operator and customers under specified service constraints;

These problems may be solved by a tree search procedure. The only difference in two models listed above is that the second model requires an integration step when the expected disutility is calculated.

1.3.4 Develop Heuristic Vehicle Routing Algorithms

It will be shown that optimal vehicle routing algorithms tend to be too computationally intensive to be used for the real-time operations in realistic traffic networks. The problem is compounded when the uncertainties of the travel times are explicitly modeled. In order to make vehicle routing algorithms operational in real-time fashion, heuristic algorithms will be developed using techniques from AI. Two types of heuristic methods will be explored. One will be using heuristic search strategies such as A* algorithm, branch pruning method and bi-directional search for the one to one shortest path search. Another approach will be to use Artificial Neural Networks (ANN) to estimate the O-D travel time distribution parameters that are used in the solution procedure of the DARP.

1.3.5 Testing

In order to evaluate and compare the performance of the algorithms proposed for both SPP and DARP, it is necessary to apply the algorithms to a set of problems and to examine the factors that impact the empirical performance. In this procedure, certain evaluation criteria such as computation time and accuracy will be specified first. The testing problems are generated as close to real world problems as possible and a network from the City of Edmonton, Alberta will mainly be used as a test bed.

1.4 ANTICIPATED CONTRIBUTIONS OF THE RESEARCH

The research is anticipated to be used in the area of transportation system modeling, operations research and the development of route guidance systems and vehicle dispatching systems. The specific contributions include:

i) The stochastic and dynamic nature of the link travel times is explicitly modeled in the same framework using probability theory. Statistical and theoretical models are developed to estimate the link travel times under the conditions of recurring and non-recurring congestion. The conclusions and models may be used by a traffic information management system that provides travel time data to the RGS and AVDS;

ii) The SPP and DARP with dynamic and stochastic travel times are analyzed and their respective solution algorithms are developed. These models and algorithms may be used in a RGS and AVDS if they are justified to be necessary;

iii) The influence of the uncertainties of link travel times in route optimization are identified using the developed models and algorithms. The conclusions may provide guidelines for the appropriate types of vehicle routing models that should be used in the RGS and AVDS;

iv) The techniques from the AI field, including heuristic search strategies and artificial neural networks are applied in vehicle routing and travel time estimation procedures. The resulting algorithms and methods may have significant applications for real-time operations of RGS and AVDS.

1.5 ORGANIZATION OF THE THESIS

This dissertation has been organized into nine chapters. Chapter 2 provides a literature review of the state of the art of the main topics related to the subject of this thesis. It includes a review of some typical link travel time representation, estimation and prediction methods used in RGS experiments around the world, a detailed discussion on various shortest path problems arising in different network models and their respective shortest path algorithms and a brief review of the k shortest path problem and algorithms. Lastly, the chapter provide an overview of the dial-a-ride problems and related issues.

In Chapter 3, the dynamic and stochastic link travel time pattern on three types of links are investigated. The first type of link represents an undersaturated, uninterrupted flow situation which prevails in most of the freeways and arterial sections excluding intersections. The second type of link represents a signal controlled link. Lastly, the third type of link represents a traffic situation with incident congestion. The link travel time distributions on these three types of links and their relationship with traffic conditions and traffic controls are discussed.

In Chapter 4, several approximation models are developed to estimate the mean and variance of the travel time of a given route in dynamic and stochastic networks. Subsequently, Chapter 5 examines the properties associated with the shortest path problem in dynamic and stochastic networks. A heuristic algorithm based on the k-shortest path algorithm is proposed and its quality of solution and computational requirements are analyzed based on a network from Edmonton, Alberta.

Chapter 6 develops various new heuristic shortest path algorithms by combining the optimal shortest path algorithms with the heuristic search strategies. The algorithmic implementation of the proposed heuristic algorithms and their computational efficiency and solution quality are discussed in detail.

In Chapter 7, the ANN are introduced as an effective method to provide a quick estimation on the O-D travel time (mean and variance). Different ANN models are developed and compared to other O-D travel time estimation methods.

Chapter 8 first discusses how the dial-a-ride system can be modeled with respect to the objectives of the system operator and customers when the O-D travel times are random variables. Two heuristic dial-a-ride vehicle routing and scheduling procedures are introduced to solve the new problem. A computational study is subsequently conducted to demonstrate the difference between the models with and without considering the dynamic and stochastic nature of the O-D travel time. The computational efficiency of the routing and scheduling algorithm with different O-D travel time estimation methods is finally demonstrated.

Lastly, Chapter 9 summarizes the main findings and conclusions of this thesis and provides the future perspective following this desertion work.

Appendix A and Appendix B discuss some theoretical estimations of the computation efficiency of two heuristic shortest path algorithms.

Appendix C provides an introduction to the ANN, with particular emphasis on the back propagation network.

REFERENCES:

Boyce, D. E., A. Kirson and J. L. Schofer, (1991), "Design and Implementation of ADVANCE: the Illinois Dynamic Navigation and Route Guidance Demonstration Program", Proceedings VNIS'91 IEEE Conference in Dearborn Michigan.

Catling, E. and P. Belchar, (1989), "Autoguide-Route Guidance in the United Kingdom", *Proceedings VNIS'89 Conference,* Toronto, Ontario.

Hall, R. W. (1986), "The Fastest Path Through a Network With Random Time-dependent Travel Times" *Transportation. Science* **20**, 182~192.

King, G. F. and T. M. Mast, (1987), "Excess Travel: Causes, Extent, and Consequences", Transportation Research Record **1111**, Transportation Research Board, Washington, D. C.

Mammano, F. and R. Summer,(1991), "Pathfinder status and Implementation Experience", Proceedings VNIS'91 IEEE Conference in Dearborn Michigan.

Rilett, L. R., (1992), *Modeling of TravTek's Dynamic Route Guidance Logic Using the Integration Model,* Ph.D. Dissertation, Queen's University, Kingston, Ontario.

Rillings, J. H. and J. W. Lewis, (1991), "TravTek", Proceedings VNIS'91 IEEE Conference in Dearborn Michigan.

Stone, J. R., A. Nalevanko and J. Tsai, (1993), "Assessment of Software for Computerized Paratransit Operations", *Transportation Research Record* **1378**, Transportation Research Board, Washington, D. C., 1~9.

Takada, K. and Y. Tanaka, (1989),"Road/Automobile Communication System(RACS) and its Economic Effect", *Proceedings VNIS'89 Conference*, Toronto, Ontario.

U. S. Department of Transportation, (1990), *Report to Congress on Intelligent Vehicle-Highway Systems*, DOT-P37-90-1.

CHAPTER 2

LITERATURE REVIEW

2.0 INTRODUCTION

The central function of an in-vehicle route guidance system (RGS) and a real-time automated vehicle dispatching system (AVDS) is the ability to provide routes that are optimal in the underlying traffic network. This has led to the necessity of modeling and estimating travel time in a traffic network and developing new solution approaches to the shortest path problems (SPP) arisen in a RGS and the dial-a-ride problem (DARP) in an AVDS. This chapter will focus on the various methods available for modeling and estimating the link travel times and O-D travel times, and the different shortest path problems and dial-a-ride problem models and the respective algorithms to these problems.

The first section of this chapter will identify some link travel time representation, estimation and prediction methods used in the primary RGS experiments around the world. The selected RGS includes TravTek and Pathfinder system in Orlando, ADVANCE system in Chicago and ALI-SCOUT system in Germany. The review of these systems will focus on how the link travel times are represented, how the link travel times are estimated from different data sources and how the link travel times during future periods are predicted.

In the second section of this chapter, the predominant shortest path problems and their associated algorithms will be discussed. The review will concentrate on the shortest path problems arising in different network models with respect to static/dynamic and deterministic/stochastic link travel costs. Various shortest path algorithms and their performance will be discussed.

In the third section an overview of the dial-a-ride problems and the related algorithms is provided. Models of static and dynamic dial-a-ride problems and algorithms are subsequently discussed.

The last section will summarize previous work in the research area which is the primary focus of this thesis. It will be shown that the current methodology has limited application for real-time application, and this will serve as the starting point of this dissertation work.

2.1 ESTIMATION AND PREDICTION OF LINK TRAVEL TIME

The primary information required for route optimization in both RGS and AVDS is the travel time on each link in the road traffic network where the RGS and AVDS would be operating. Link travel times can be obtained from a variety of sources such as loop detectors, probe vehicles, traffic simulation models and et al. From these data sources both historical and current values of the link travel time can be generated. A prediction model is then required to forecast the link travel time during future time period. These procedures are called link travel time data fusion and link travel time prediction respectively. Various data fusion methods and travel time forecasting methods have been proposed during the development of the RGS demonstration projects. This section provides a general review of link travel time representation, data fusion procedures, link travel time forecasting methods used in RGS demonstration projects: ADVANCE, ALI-SCOUT, TravTek and Pathfinder.

2.1.1 ADVANCE System

The ADVANCE project was initiated in Chicago in 1992. The RGS in the ADVANCE system is a vehicle-based or distributed route guidance system, e.g., each equipped vehicle has an on-board computer which calculates the "best" route for the driver. The concept of the route guidance implemented in ADVANCE RGS is dynamic in the sense that the route optimization is based on anticipated link travel times. The final method used to predict the link travel times is still under development and the methods discussed in the following paragraphs are only used in the initial deployment of the ADVANCE system (Boyce et al., 1993; Tarko, A. et al. , 1993).

Two types of link travel time profiles are used: static and dynamic. The static profiles, kept in each equipped vehicle, represent the historical behavior of the link traffic and are used for route calculation when the road traffic is considered in a stable condition. In ADVANCE, this is defined as a non-incident situation. Dynamic profiles are used by the traffic information center (TIC) to update the static profile when the real-time link travel time is detected to deviate significantly from the historical value. Based on the real-time link travel time, the future link travel times (for example, 40 minutes), are estimated and the differences between the anticipated link travel time with the historical travel time are sent to each equipped vehicle for the route calculation.

Two types of data source are used for the estimation of link travel time during the current time period. One data source is detectors embedded in the road network. The output of a detector is the real-time occupancy and volume on the link. The expected detected link travel time (EDTT) is derived from the detected occupancy based on a link specified regression model. Another data source is the RGS-equipped vehicles that provide the time to take them to travel on each link (EPTT) along their route. These two data sources are fused to generate expected on-line link travel time (EOTT) based on Bayes's inference rule:

$$\mathrm{EOTT} = \frac{\mathrm{EDTT} / \sigma_d^2 + \mathrm{EPTT} / \sigma_p^2}{1 / \sigma_d^2 + 1 / \sigma_p^2} \qquad (2\text{-}1)$$

Where σ_d^2 is the variance of the detected link travel time; σ_p^2 is the variance of the link travel time from probe vehicles. This on-line travel time is then fused with historical travel time to generate new historical link travel time using similar formula as 2-1.

If the value of EOTT does not deviate significantly from the historical link travel time during the same time period, the historical link travel time profile would be used for the prediction of future link travel time. Otherwise, a new dynamic link travel time profile would need to be created. Although the link travel time prediction method used in ADVANCE experiment has not been discussed in the recent literature (Boyce et al., 1993), a conceptualized approach to forecasting the link travel time has been proposed by Chen and Underwood (1991). The approach begins by simulating the routing of vehicles through the network to obtain the dynamic link travel time profile, then rerouting the vehicles based on this simulated dynamic link travel time profile, and iterating this process until it converges to a steady state. Obviously, to be effective, the simulation procedure must be significantly faster than real time. This approach is at the development stage in respect of both theory (dynamic traffic assignment) and practice (need powerful computers that parallel computation).

It should be noted that the link travel time estimation models proposed for the ADVANCE project inexplicably assumed that the link travel time is dynamic and stochastic (e.g., the variance of the link travel time is used in Equation (2-1)). However, the actual information used in the vehicle routing process is the average link travel time. That is, the stochastic attributes of the link travel times are not considered in the route optimization. As part of the ADVANCE operation test, Rouphail (1995) provided some theoretical and empirical evidence on the variation and distribution of the link travel times. They proposed some microscope travel time distribution models for vehicle entering a signalized traffic link as a function of traffic flow and traffic control. Their research also shows that the variation of link travel time with traffic control is very significant and the probability distribution is bimodal.

2.1.2 ALI_SCOUT System

The ALI_SCOUT system which began its field test in Berlin in 1988, is a centralized route guidance system where the routes are calculated by a central computer and then sent to the vehicles through road side beacons locating at mojor intersections. The route calculation in ALI-SCOUT is based on anticipated travel time on links. The link travel time prediction method was proposed by Hoffman and Janko (1990) and includes a method for creating an historical link travel time profile and a dynamic prediction algorithm.

A historical travel time data base is maintained for each link. The data base is composed of a standard profile for each link. This standard profile is updated daily using the data from the preceding day. If $t_{ij.n}$ represent the average travel time during nth time period for link (i,j), then,

$$\text{new } t_{ij.n} = \alpha t'_{ij.n} + (1-\alpha) \text{ old } t_{ij.n} \tag{2-2}$$

Where $t'_{ij.n}$ is the new observation of the average travel time of the nth period for link (i,j); α is a weighting coefficient.

The travel time prediction method assumes that the ratio of current travel time and the mean travel time for the historical data base remains unchanged over future time period. This ratio is called deviation coefficient ($\rho_{ij.k}$) and defined as follows, if the current time period is k:

$$\rho_{ij.k} = \frac{t_{ij.k}}{\tilde{t}_{ij.k}} \tag{2-3}$$

Where $t_{ij.k}$ is the mean travel time from standard profile; $t_{ij.k}$ is the current average travel time on link (i,j) for time period k. If there is no current travel time available, then $\rho_{ij.k}$ is set to 1. The coefficient can be smoothed by combining it with the deviation coefficients of preceding periods or of the neighboring links.

Finally, the predicted travel time for link (i,j) at a future time interval m, $t^*_{ij.m}$, is then given by:

$$t^*_{ij.m} = \frac{t_{ij.m}}{\rho_{ij.k}} \qquad (2\text{-}4)$$

Koutsopoulos and Xu (1993) proposed an information discounting strategy as an extension of the prediction method discussed above. Instead of assuming the deviation coefficient remains constant for the entire prediction horizon, they predict the travel time on link (i,j) at a future time period m, $t_{ij.m}$, by discounting the travel time $t^*_{ij.m}$ from Equation (2-4) to both the travel time from origin node s to node i, P_{si} and the standard deviation $\sigma_{ij.m}$ with an exponential function as follows:

$$\check{t}_{ij.m} = t_{ij.m} + e^{-\theta\sigma_{ij.m}P_{si}}\left(t^*_{ij.m} - t_{ij.m}\right) \qquad (2\text{-}5)$$

Where m is the future time period when the vehicle arrives at node i; $t_{ij,m}$ is the historical travel time for link (i,j) at time period m; θ is a constant scalar, which can be adjusted to provide better fit between projected and actual travel time. A similar method was also proposed by Rilett (1992).

It should be noted that link travel times in these methods are considered as random variables and a smoothing process is used to obtain the average profile of the link travel times. However, the variation of the link travel time is not considered in the path finding procedure.

2.1.3 Pathfinder and TravTek Systems

The Pathfinder in Los Angeles, California, and TravTek, in Orlando, Florida, represent the first two implementations of ITS in the North America. The current implementations of both systems have similar architecture with a central computer that gathers data from variety of sources, generates and disseminates the *prevailing status* of the network traffic to the equipped vehicles. The prevailing status of the network traffic is represented by the current link travel time in the road network. In the Pathfinder system, the current link travel time is transferred into congestion levels for display in the in-vehicle computer screen. In the TravTek system, the current link travel time is used to calculate the optimal route from current position to the destination. The generation of current link travel times, i.e., data fusion, is based on data from various sources.

The data fusion method proposed for Pathfinder and TravTek involves a fuzzy logic maximum height solution process (Sumner 1991). The data sources considered includes probe vehicles, loop detector, TRANSYT modeling, operator interface and historical files. Each data source is represented by two fuzzy variables including quality of data and aging. A score is produced by allocating the quality number to the data source and then linearly decrement the score each minute using the age factor. In any minute, the source with the maximum score is considered the best one and the data from that source are used to estimate the link travel time. The problem with the fuzzy logic method is that only one data source is actually used in link travel time estimation. This means that the data from the remaining sources are completely neglected or ignored when estimating the current travel time.

2.2 SHORTEST PATH PROBLEM AND ALGORITHMS

The key component in any RGS is the method used to identify the "best" path between two points, or shortest path algorithm. The shortest path algorithm is also an important part in any AVDS where an accurate estimation of the minimum O-D travel time and/or cost is required in the vehicle routing and scheduling process. The objective of a shortest path algorithm is to find the path with minimal travel cost from an origin location to a destination location. The travel cost is a general term representing one or a combination of travel time, travel fee and safety level et al. In this research, travel time will be used without any further notation although the results are applicable for any generalized cost. This shortest path problem has been studied for over thirty years in diverse fields such as computer science, communication and transportation engineering. Numerous algorithms and extensions have been proposed. The review will focus on the basic shortest path problems, algorithms and their extensions including dynamic and/or stochastic shortest path problems.

In order to simplify the description of algorithms, an annotation is provided as follows. A road traffic network is represented by a digraph G(N,A) consisting of a set of nodes N and a set of arcs A (or links used in this paper). Denote the number of nodes |N|=n and the number of links |A|=m. A link $a=(i,j) \in A$ is directed from node i to node j and associated to a travel time τ_{ij}. A path from an origin(s) to destination(d) is a sequential of nodes and links: s, (s,j), (j,.........,(i,d), d. The travel time of the path is the sum of the times along all its links.

The following notation will also be used.

$L_{(i)}$ = the minimal travel time to node i, starting at origin node s. It is often known as the label of the node;

$P_{(i)}$ = the preceding link (a pointer) on shortest path to node i;

Q = a list of nodes placed in a certain order for examination;

2.2.1 Shortest Path in a Static and Deterministic Network

The static and deterministic shortest path problem is defined as finding a minimum time path in a network where the travel time associated with each link is a constant. Without any other constraints, this problem can be solved in polynomial time. Due to its computation tractability, most of the research in this area has focused on an optimal solution of the problem and how to improve the efficiency of the optimal algorithms by using different data structures and search strategies.

An optimal shortest path algorithm is essentially an application of dynamic programming theory to the search of shortest path in a graph. The shortest path is found through an iterative decision making procedure from the origin node (s) (or destination node) to destination node (or origin node) by applying following recursive formula:

$$L_{(j)} = \min_{i \neq j}\{L_{(i)} + \tau_{ij}\} \ ; \quad L_{(s)} = 0 \tag{2-6}$$

This dynamic programming problem can be solved by effective labeling algorithms which have following common procedure (Assume that the search starts from origin node s):

```
Step 1: Initialization:   $i = s; L_{(i)} = 0 ; L_{(j)} = \infty \ \forall j \neq i ;$
                          $P_{(i)}$ = NULL
                          Define the scan eligible node set $Q=\{i\}$;
Step 2: Stop Rule:        IF $Q=\varnothing$ THEN stop.
                          ELSE select and remove a node i from Q;
Step 3: Node Expansion:   Scan the forward star of the node i. For each link a=(i,j)
                          IF    $L_{(i)} + c_{ij} < L_{(j)}$ ,
                          THEN  $L_{(j)} = L_{(i)} + \tau_{ij}$ ; $P_{(j)} = a$ ; insert node j into Q;
Step 4: Iteration:        GOTO step 2.
```

The major variations between different algorithms pertain to the data structure used to form the *scan eligible node set* and the manner in which the nodes are identified and selected for examination. Based on the behavior of an algorithm, the optimal shortest path algorithms are usually classified as two categories: label correcting and label setting algorithms. The 'label' refers to the travel time.

2.2.1.1 Label correcting algorithms:

The label correcting algorithms use a list structure to manage the scan eligible node sets that need to be examined during the shortest path tree building process. Variations of the list operation policy make up of different label correcting algorithms such as depth first search (last-in-first-out list), breadth first search (first-in-first-out list) and derived search strategies.

There are three typical algorithms in this category:

(1). Label correcting with queue (Moore, 1969) : In this algorithm, the addition of new node is only allowed at tail and deletion only at head. Its computation complexity is O(m.n);

(2). Label correcting with double ended queue (Pape, 1974) : In this algorithm, the addition and deletion are possible at either end, depending on whether or not the node has been examined before. This type of list operation reduces the probability that a selected node will have to be reanalyzed at some future point and consequently improves the overall search efficiency. However, its worse case complexity is $O(n.2^n)$, that is, it is worse than label correcting with queue algorithm.

(3). Label correcting with threshold lists (Glover *et al.*, 1985) : The development of this algorithm was an attempt to combine the advantages of both label setting and label correcting algorithms. In this algorithm, two lists called NOW and NEXT are maintained during the shortest path search. At each iteration, a node in NOW list is selected for examination in a LIFO manner. When the NOW list is exhausted, a threshold value is calculated to determine which nodes currently in NEXT may be moved to NOW list. The expected efficiency of this algorithm results from both avoiding the necessity of maintaining a complete ordered list as in label setting algorithms and reducing the redundant calculation as in label correcting algorithms. The computation complexity of this algorithm is same with label correcting with double ended queue, i.e., $O(n.2^n)$.

The major feature of the label correcting algorithms is that it cannot identify the optimal shortest path between the root node with another node before they explore every node in the network. This attribute makes them suitable in the situation when all the shortest paths from the root node to the other nodes in the network need to be identified.

2.2.1.2 Label setting algorithms:

In label setting algorithms, a scan eligible node set is maintained orderly based on their current path time from the root node, i.e., their labels. During the shortest path search the node with least label is selected for examination and at same time the shortest path to this node is identified. The major difference among the label setting algorithms is the data structure used to maintain the ordered node set. There are following three types of label setting algorithms popularly identified:

(1). Label setting with sorted list (Dijkstra, 1959): sorted list is the simplest way to store the ordered scan eligible node set. The node with minimum label is at the head of the list. Insertion of a new node takes $O(n)$, that implies an $O(n^2)$ computation complexity;

(2). Label setting with binary heap (Tarjan, 1983): in this algorithm, a binary tree data structure called heap is used to store the ordered scan eligible node set. This tree data structure reduce the computation complexity into O(n.log n) from O(n^2) required by using sorted list;

(3) Label setting with buckets (Dial, 1969): in this algorithm, a pointer array is used to store the ordered buckets . Each bucket in the array corresponds to certain label range and stores the nodes with its label within the range. It therefore requires a pointer array with enough length to handle the maximum possible label (or path length). This algorithm can reduce the computation efforts to select a node with minimum label and insert a node in the scan eligible node set. However, its sensitivity to the network size and link length makes it inefficient for large network problems.

2.2.1.3 Computation studies on label correcting and label setting algorithms:

It has been found insufficient to use the worse case complexity to assess the ability of a particular algorithm to perform certain functions. This is especially true when the problem has special structure that does not fit the general situation. Many comprehensive computation studies on the performance of the shortest path algorithms have therefore been conducted in different research fields. Most of the comparisons are based on criteria such as computation time, complexity of implementation and storage requirements. According to the literature that focuses on transportation networks (Gallo et al., 1984; Hung et al. 1988; Vuren et al., 1988), the following conclusions may be summarized. Among the label correcting algorithms, the label correcting algorithm with the double ended queue and the label correcting algorithm with threshold lists are found dominant.

The difference in computational efficiency between those two label correcting algorithms are trivial in transportation road networks. However the former is much easier to implement and consequently is more widely used algorithm. On other hand, the label setting algorithm with binary heap is one of the fastest algorithm among label setting algorithms. In addition, the label correcting algorithm is always faster than the label setting algorithms. The last conclusion may result from comparison conditions. It should be noted that all comparison studies found in the literature are based on one-to-all mode, i.e., finding the shortest path from a root node to all the other nodes in graph. If only the shortest path between a pair of nodes is required as in a RGS, label setting algorithms could be better than label correcting algorithms.

2.2.2 Shortest Path in a Dynamic Network

An important extension of static shortest path problem is the dynamic shortest path problem. This problem entails finding the minimum time path in a network where the travel time on some or all links is dynamic, that is, the link travel time changes with time of day. This problem is especially important in road traffic networks where recurrent congestion is a common phenomenon.

The dynamic shortest path problem was first studied by Cooke and Halsey (1966) and further described in Drefus's paper (1969). They concluded that the problem can be solved by using Dijkstra's algorithm and the algorithm is as efficient as if the link travel time is not dynami. Kaufman and Smith (1990) first proposed the application of the dynamic shortest path algorithm in a RGS and further clarified the sufficient condition that ensure that the optimal solution can be found.

Assume that the link travel time is $\tau_{ij(T_i)}$ with T_i representing the entry time on link (i,j). The following new recursive formula can be established:

$$L_{(j)} = \min_{i \neq j}\{L_{(i)} + \tau_{ij(T_i)}\} \; ; \quad L_{(s)} = 0 \; ; \; T_s = \text{given} \qquad (2\text{-}7)$$

Where $L_{(i)}$ and $L_{(j)}$ are respectively the travel time from origin node s to node i and to node j; T_s is the departure time at the origin node s. If the label associated with node j has a minimum value when the path goes through node i = i*, then the arrival time (or departure time) at node j can be updated with the following formula:

$$T_j = T_{i*} + \tau_{i*j}(T_{i*}) \qquad (2\text{-}8)$$

It is obvious that it will not impose extra computation burden to solve the above recursive formula by using labeling algorithms presented in section 2. 2. 1. The only difference is that departure time at each node is a new decision variable and must be updated at each recursive step. However, because the link travel time is a function of the entry time at the link, the labeling algorithms are not allowed to start from the destination node and work backward to find the shortest path as in a static network.

2.2.3 Shortest Path in a Stochastic Network

In many application situations, the link travel time in a network is not deterministic and is governed by some discrete or continuous stochastic process. This is typical in a road traffic network where the travel time on each road segment may be considered as a random variable, resulting from many factors such as the fluctuation of traffic demand and variation in individual driver's behavior. The shortest path between two nodes in a stochastic network depends on the state of the network and therefore is not necessarily always composed of the same links. Contrast to the deterministic shortest path problem, the shortest path problem in stochastic networks can have many different models. Examples include how to find the probability distribution function of the shortest path length in a stochastic network and how to find the path that stochasticly dominates over all the other paths based on a specific utility funtion.

The earliest work in this area is attributed to Frank (1968) who identified a method of determining the probability distribution of the shortest path length. In that paper the links in a network are assumed to have independently distributed length (travel time) with a continuous probability function. Mirchandani (1976) extended Frank's work by considering the case that the probability function of the link length is discrete. Although these work provided insight of the characteristics of the shortest path length between two nodes, they cannot be applied to find the optimal path.

2.2.4 Shortest Path in a Dynamic and Stochastic Network

A dynamic and stochastic network has the combined attributes of both dynamic and stochastic networks. That is, the link travel times in general manifest time-dependent pattern (dynamic) , and are not deterministic at any time moment or time interval. Dynamic and stochastic networks are the most feasible models of road traffic network where the travel time on each road depends on the time of day (e.g., peak and off-peak period), and at same time has certain amount of variation at any time period.

The shortest path problem in dynamic and stochastic networks was first studied by Hall (1986) which still remains the only research work on this specific problem. Hall's work focused on the problem of finding the path that has the minimum expected travel time from an origin node to a destination node. That paper first demonstrated the augment that the traditional label setting or correcting algorithms (discussed in section 2.2.1) may fail to find the expected shortest path. A new algorithm was then proposed as an attempt to handle the dynamic and stochastic attribute of the network. The algorithm is fundamentally an enumeration procedure based on a k-shortest path algorithm. The algorithm processes as follows:

Step 1: **Initialization:** Set $k = 1$; $\tau_u = \infty$;

Find the (1st) shortest path from node s (origin) to destination node d, based on minimum possible travel times over links in the network. Call the path P_1, set τ_l equal to the minimum possible travel time over P_1 and calculate the corresponding expected travel time T_1

Step 2: **Stop Rule:** If $T_k < \tau_u$: $\tau_u = T_k$

$P = P_k$

If $\tau_u < \tau_l$: P is the optimal path

τ_u is the minimum expected travel time

Stop

Step 3: **Expansion:** Set $k = k+1$

Based on the minimum possible travel times on each link, find the kth shortest path from node s to the destination node, call it P_k.

Set τ_l equal to minimum possible travel time over P_k, calculate the expected travel time T_k over P_k

Step 4: **Iteration:** Goto Step 2

This algorithm provides a method to exploit the expected shortest path in dynamic and stochastic networks, however, there are some related issues that need to be addressed before it can be applied in real applications.

First, Hall's paper has not discussed about how to calculate the expected travel time of a given path when the link travel times on the path are dynamic and stochastic. This computation may be simple for the network which has small number of nodes and links, and the random link travel time has small number of states. However, it will not be trivial for large scale networks.

The second issue is related to the computation efficiency of the algorithm. Because there is no computation experiment and complexity estimation conducted on this algorithm, it needs to be verified whether or not it can be applied to real world road traffic network. This doubt arises from the conjecture that the algorithm may have to exploit a large number of k-shortest paths before it finds the expected shortest path.

The last issue is that Hall's augment is based on a transit network, it is not necessarily the case for the road traffic networks because these two types of networks have very different link travel time patterns. It is therefore necessary to further investigate how much estimation error would be induced if the standard shortest path algorithms are used.

2.3 K SHORTEST PATH PROBLEMS AND ALGORITHMS

The above section described the problem of finding the shortest paths between two nodes in various types of network model. In many situations, however, there is a need to find the nearly optimal paths such as the second shortest and the third shortest paths (instead of only the shortest paths). The underlying problem is often referred as to k shortest path problem (k-SPP), and a solution procedure to k-SPP is called k shortest path algorithm. It should be noted that the k-SPP assumes that the underlying network is deterministic.

As a typical example, k shortest path algorithms have been exclusively used to solve the constrained shortest path problem which is to find the shortest paths that must satisfy a set of constraints. In this thesis the k shortest path algorithms will be used in a proposed solution approach to the DSSPP as discussed in Chapter 5.

There are generally three classes of k shortest path algorithms. The first one is Dreyfus's algorithm (Dreyfus, 1969). The second class is due to Shier, representing a generalization of the labeling shortest path algorithms discussed in Section 2.2.1. The last one is the algorithms based on path deletion concepts (Azevedo *et. al.*, 1993). Because this thesis will not examine the efficiency of various k shortest path algorithms, Shier's algorithms will be selected for use for its close relation with the shortest path algorithms. The following section provides a detailed discuss on Shier's algorithm.

As the labeling algorithms for the shortest path problem, the k shortest path labeling algorithms can also be classified as label correcting and label setting algorithms. The concepts and characteristics of these two algorithms are similar to the shortest path labeling algorithm. The following graph will present the k shortest path label setting algorithm which has been adapted in this thesis.

Using the same notations for the SPP presented in Section 2.2 except that a k vector of labels $L_j=\{\ell^1_j, \ell^2_j, \ldots, \ell^k_j\}$ is assigned to every node j, where the entry ℓ^i_j of L_j represents the current label of i*th* shortest path to node j. A vector q is used to store the minimal temporary entry for each node. The k shortest path label setting algorithms proceed sequentially and at each step identify a new correct entry of some k vector of some node in the final solution. This process continues until all the component value of the k vector L_g corresponding the destination node g are made permanent. The following lists the procedure of finding the k shortest paths from origin node (s) to destination node (g) (the search starts from the origin node).

```
Step 1: Initialization: $i = s$; $L_{(i)} = \{0, \infty, \infty, \ldots \infty\}$; $q_{(i)} = 0$;
                        $L_{(j)} = \{\infty, \infty, \infty, \ldots \infty\}$; $q_{(i)} = \infty \ \forall j \neq i$;
                        Define the scan eligible node set $Q = \{i\}$;
Step 2: Stop Rule:      IF $Q = \varnothing$ THEN stop.
                        ELSE select the node i with smallest temporary entry q value from
                        Q. Assume the sequence of this entry in $L_i$ is $k^*$, then
                        IF $k^* = k$ THEN
                                IF $i = g$ THEN stop
                                ELSE remove node i from Q
                        ELSE $q_{(i)} = \ell^{k^*+1}_j$ and insert node i into Q;
Step 3: Node Expansion: Scan the forward star of the node i. For each link $a = (i,j)$
                        IF $q_{(i)} + c_{ij} < \ell^{k^*}_j$ THEN $\ell^{k^*}_j = q_{(i)} + c_{ij}$;
                        IF $\ell^{k^*}_j < q_{(j)}$ THEN $q_{(j)} = \ell^{k^*}_j$ and insert node j into Q;
Step 4: Iteration:      GOTO step 2.
```

It should be noted that this procedure only identifies the k shortest path length from the origin node to the destination node, the actual k shortest paths are found through a backtracking procedure based on the k vector value of each node.

2.4 DIAL-A-RIDE PROBLEMS AND SOLUTION METHODS

There has been considerable research on the dial-a-ride problem (DARP) in operations research and transportation science over the past 30 years. The interest on this problem is mainly attributed to the development of paratransit system where the daily operation requires the routing and scheduling of vehicles for the handicapped, the elderly and other people who cannot access the fixed route public transit system. In the dial-a-ride system, customers call a dispatcher in order to request service. Each customer specifies a distinct pick-up and delivery location in the service area and usually, a desired time for pick-up or drop-off. The problem is to develop a set of "optimal" routes and schedules for vehicles to carry the customers from their pick-up locations to their drop-off locations. The DARP is a constrained version of the Vehicle Routing Problem (VRP), the constraints relate to the precedence relationships between the origin location and destination location of each customer (Bodin et al., 1983). A more recent survey of the literature was provide by Savelsbergh and Sol (1995).

The DARP is traditionally classified into two categories based on the characteristics of customers' service requests: static and dynamic. In the static DARP, all the customers reserve service in advance, e.g., one day ahead, so that complete information about the customers is known before the routing and scheduling is carried out. This problem is also called advance request DARP. On the other hand, if some of the customers request immediate service, then the routing and scheduling are done in real time and the problem is referred as to the dynamic or the demand responsive DARP. In the dynamic DARP, the customers requesting immediate service must be inserted into the existing route.

The past research on the DARP mainly focused on how to realistically model the operation scenario (operator's and customers' requirements), and how to solve large scale

DARP for most real operation situations . The section 2.3.1 outlined the research work done on static DARP. The dynamic DARP and algorithms are reviewed in section 2. 3. 2.

2.4.1 Static Dial-A-Ride Problem and Algorithms

The static DARP has further sub-classifications based on the number of vehicles used (i.e., single vehicle vs. multiple vehicles), and any service requirements (i.e., with or without time windows).

The single vehicle static DARP was first studied by Psaraftis (1983). The objective used is a linear combination of total route duration (representing the operator's disutility) and total waiting time and riding time of all customers (customers' disutility). The constraints on the problem include vehicle capacity and maximum position shift for each customer between his position in the reservation list and the position in the sequence of pick-up. A dynamic programming algorithm was developed to solve the problem with and without time windows optimally. The optimal algorithm requires $O(N^2 3^N)$ time and is only tractable for small size problems (less than 10 customers).

Sexton and Bodin (1985a,1985b) proposed an approach for the single vehicle static DARP with time windows. The operation scenario modeled is characterized as that, each customer has a desired drop-off time. The objective used is to minimize the total inconvenience which a customer may experience. The total inconvenience is expressed as a linear combination of the excess ride time and the deviation from the desired drop-off time. Based on their proposed formulation, the problem is solved through an iterative procedure which alternately finds the route and schedule.

In contrast to the single vehicle case, the multiple vehicle DARP is a more realistic model of most applications and has received a significant amount of attention. Many heuristic approaches to this problem have been developed and most of these approaches are alike in terms of algorithmic philosophy. The algorithm proposed by Jaw (1986) is discussed below.

Jaw's heuristic algorithm derived some concepts from the work done by Wilson et al. (1977) for the dynamic DARP. (The dynamic DARP is discussed in next section) . The algorithm is composed of a search for the feasible insertion of customers into the work schedule and an optimization step. The feasibility of inserting a customer is verified on the basis of the following assumptions on the operating scenario:

1. Each customer specifies either a desired pick-up time (DPT) or a desired drop-off time (DDT). No DPT- (DDT-) specified customer will be picked up earlier than his/her DPT (DDT);
2. No customer's actual ride time will exceed a given maximum;
3. The time deviation between the actual pick-up (drop-off) time and the desired pick-up (drop-off) time of a customer will not exceed a given maximum value for DPT-specified (DDT-specified) customers.

With these constrains all the customers are sequentially inserted into the work schedule of each vehicle. At each step, the customer that generates the least extra COST when it is inserted into a feasible position, is selected as the best insertion. The COST is defined as a weighted sum of disutility of all the customers and of operator costs. The disutility of a customer (DU_i for customer i) is defined as

$$DU_i = DU_i^d + DU_i^r \tag{2-9}$$

Where:

DU_i^d = disutility due to deviation from most desired time

= $C_1 x_i + C_2 x_i^2$

and

DU_i^r = disutility due to excess ride time

= $C_3 y_i + C_4 y_i^2$

Where C_1, C_2, C_3 and C_4 are parameters that can be adjusted to reflect customers' preference pattern on the deviation from their desired times (x_i) and excess ride times (y_i).

The incremental cost, VC, to the system operator resulted from insertion of a new customer (customer i) is defined as a combination of additional vehicle travel time (z_i) and the change of vehicle slack time (w_i).

$$VC_i = C_5 z_i + C_6 w_i + U_i (C_7 z_i + C_8 w_i)$$

Where C_5, C_6, C_7 and C_8 are externally set parameters; U_i is an indicator of system work load defined as a ratio of the number of customers to be serviced to the number of vehicles available. With this parameter, the general objective function will place more emphasis on the system operator's cost when the service demand is heavy or vehicle resources are scarce.

2.4.2 Dynamic Dial-A-Ride Problem and Algorithms

In contrast to a static dial-a-ride system, a dynamic dial-a-ride system accepts new customers who call to request immediate service. It is therefore required for the system to have the function to immediately determine the assignment of a new customer to a vehicle and the new route and schedule for the vehicle that the customer is assigned to. At the time of a new request, each vehicle in the system is on his/her way to pick up or drop off a customer based on his/her earlier assigned route and schedule. Some of the earlier customers who have been already delivered to their destinations, are no longer considered in this problem. The other earlier customers are either on board to be delivered to their destinations or are waiting to be picked up.

On the dynamic DARP, most of the work has been done by Wilson and his colleagues at MIT (1976,1977). The approaches developed by them have been extensively tested in the Rochester, New York Dial-A-Ride Demonstration Project, which is also the earliest computerized dial-a-ride demand responsive system in North America. Although the initial system has not been expanded because of the higher cost of the computer resource at that time, these algorithms are still feasible.

The Wilson's approach is an insertion algorithm. The algorithm examines all possible insertion of pick-up and drop-off stops for the new customer into the routes for all available vehicles and selects the best way to incorporate the new customer into the existing routes and schedules. In order to select the best vehicle and positions to assign the new customer, an objective function is defined as a combination of the incremental disutility of system customers' and incremental cost of the system operator after the new customer is inserted.

The disutility of customer i, D_i, is defined as

$$D_i = a\, w_i^2 + b\, R_i^2 + c\, P_i^2 \qquad (2\text{-}10)$$

where, a, b and c are parameters which can be adjusted for different type of customers depending on the service demanded; w_i is the desired pickup time for customer i; P_i is the scheduled pickup time for customer i; R_i is the ride time from pick-up to drop-off for customer i; Therefore the total incremental disutility for all customers after the new customer is inserted can be calculated. For the inserted customer there is only the disutility after he/she has been inserted into an existing route.

The objective function related to system operator attempts to spread the tour length equally among all the vehicles as well as minimize the total travel time. This term, VC_k for vehicle k, is defined as:

$$VC_k = (L_k^{new} - L_k^{old})(d \cdot L^{ave} + e \cdot L_k^{old}) \qquad (2\text{-}11)$$

Where d and e are parameters; L_k^{new} is the tour length for the vehicle under consideration after the new customer is inserted into the vehicle; L_k^{old} is the tour length for the vehicle under consideration before the new customer is inserted into the vehicle. L^{ave} is the average tour length of all the vehicle after assignment.

2.5 SUMMARY

This chapter has provided an overview of the research literature on the estimation and prediction of link travel times, the shortest path problems and dial-a-ride problems. The main points are summarized as follows.

1. On the estimation and prediction of link travel time in the road traffic network

    - The link travel time has been implicitly considered as a random variable and represented by its mean and variance. However the mean travel time is exclusively used in route optimization in the existing RGS experiments. The variance is only used as an assistant parameter for data fusion and prediction of travel times.

    - The prediction of link travel time still resorts to some simple heuristic algorithms. The methodology of link travel time prediction by using dynamic assignment procedure is still under development, and the computation feasibility of its implementation needs to be further investigated.

- The link travel time is commonly assumed to be normally distributed. Little work has been done on how to model the stochasticity of the link travel time, and how to estimate and predict the distribution parameters or moments of the link travel time, instead of just mean travel time.

2. On the shortest path problem

   - Among the optimal algorithms, label setting algorithm with binary heap is found to be one of the best to find the shortest path between two specified locations while the label correcting algorithm with double ended queue is preferred for the case to find the shortest paths from one location to many other locations.

   - The optimal shortest path algorithms, such as label setting algorithms and label correcting algorithms, tend to be too computationally intensive for real-time one-to-one applications in realistic traffic networks. More efficient shortest path algorithms such as heuristic algorithms need to be developed;

   - The shortest path problem in a dynamic network can be solved as efficiently as the static shortest path problems by the labeling algorithms. However it is not the case when the network is both dynamic and stochastic. Further study is necessary to solve this problem;

- Under the network with stochastic link travel time, the variation of path time could become another important criterion for route selection. However the resulted problems have not been well studied;

3. On the dial-a-ride problem

- Operation researchers and practitioners have been trying to model the dial-a-ride problems more realistically by properly considering both the system operator's cost and the system customers' requirement in the route and schedule optimization. However all of the models assume that the travel time between two locations, or O-D travel time, is static and deterministic;
- Solution methodologies to the DARP are still dominated by heuristic algorithms, mostly of the insertion type;

REFERENCE:

Azevedo, A. J., S. Costa, S. Madeira and E. Q. V. Martins (1993), "An algorithm for the ranking of shortest paths,". European J. of Operations Research 69 , 97~106.

Bodin, L., B. Golden, A. Assad and M. Ball (1983), "Routing and Scheduling of Vehicles and Crews. The State of Art ,". Computers and Operations Research 10 , 69~211.

Boyce, D., Rouphail, N., and A. Kirson, (1993), "Estimation and Measurement of Link Travel Times in the ADVANCE Project", IEEE-IEE Vehicle Navigation & Information Systems Conference, Ottawa - VINS'93.

Chen, K. and Underwood, S. E. (1991), “Research on Anticipatory Route Guidance” Proceedings of The Second Vehicle Information and Navigation System Conference, October 20~23, 1991, Dearborn, MI, Society of Automatic Engineering, Vol. 1, 427~440.

Christophides, N. (1976), "Worst-case Analysis of a New Heuristic for the Travel Salesman Problem" Report 388, Graduate School of Industrial Administration, Carnegie Mellon University.

Cooke, L. K. and E. Halsey, (1966), "The Shortest Route through a Network with Time-Dependent Inter-modal Transit Times" Journal of Mathematical Annals and Application 14, 493~498

Dial, B. R, (1969), "Algorithm 360: Shortest Path Forest With Topological Ordering" Communs Ass. Comput. Mach. 12, pp. 632~633.

Dijkstra, E., (1959), "A Note on two Problems in Connection with Graphs," Numerical Mathematics 1.

Dreyfus, S. E., (1969), "An Appraisal of Some Shortest Path Algorithms," Operations. Research. 17, 395~412 .

Frank, H. (1968), "Shortest Path In Probabilistic Graphs", Operations Research. 17, 583~599.

Gallo, G., and S. Pallottino, (1984), "Shortest Path Methods In Transportation Models," In Transportation Planning models by M. Florian (editor), 227~256.

Glover, F., D. Klinkgman and N. Philips, (1985), "A new Polynomially bounded Shortest Path Algorithm," Operations. Research., Vol. 33, 65~73.

Golden, B. L. and M. Ball, (1978), "Shortest Paths with Euclidean Distance: An Explanatory Model" Networks, Vol. 8, 297~314.

Grubba, Hangen, and Knuckeart (1991), "Implementing the FAST_TRAC ATMS/ATIS Demonstration Program" VNIS conference 1991, DEARborn, MI.

Hall, R. W. (1986), "The Fastest Path Through a Network With Random Time-dependent Travel Times" Transportation. Science. 20, 182~192.

Hart, P. E., N. J. Nilsson, and B. Raphael, (1968), "A Formal Basis for the Heuristic Determination of Minimum Cost Paths" IEE Trans. System Science and Cybernetics, Vol. SSC-4, No. 2, pp. 100-107.

Hoffman C. and Janko J. (1990), "Travel Time as a Basic of the LISB Guidance Strategy" Paper presented at the IEEE road Traffic Control Conference. London.

Hung, M. S. and J. J. Divoky, (1988), "A Computational Study of Efficient Shortest Path Algorithms," Computer Operations. Research. Vol. 15, No. 6, 567~576.

Jaw, J. Odoni, A. R. , Psaraftis, H. N. and N. H. M. Wilson, (1986), "A Heuristic Algorithm For The Multi-vehicle Advance Request Dial-A-Ride Problem With Time Windows", Transportation. Research. -B , Vol. 20B, No. 3, pp. 243-257.

Kaufman E., J. Lee and R. L. Smith, (1990), "Anticipatory Traffic Modeling and Route Guidance in Intelligent Vehicle-Highway Systems," IVHS Technical Report-90-2, University of Michigan.

Koutsopoulos, H. N. and Xu, H. (1993), "An Information Discounting Routing Strategy For Advanced Travel Information Systems", Transportation Research, Part C. Vol. 1, No. 3, pp. 249~264.

Kuznetsov, T., (1993), "High Performance Routing for IVHS," IVHS America 3rd Annual Meeting, Washington, D. C.

Lin S. and B. Kernighan, (1973), "An Efficient Heuristic Algorithm for the Travel Salesman Problem" Operations Research 21, 489~516.

Mirchandani, P. B. ,(1976), "Shortest distance reliability of probabilistic networks", Computer. Operations Research. 3 , 347~356.

Moore, E. F, (1959), "The Shortest Path Through A Maze" In Proceedings of the International Symposium on Theory of Switching, Harvard University Press, Cambridge, Mass., pp. 285~292.

Nicholson, T. A. J., (1966), "Finding the Shortest Route Between Two Points in a Network," Computer Journal 9, 275~280.

Nilsson, N. J., *Problem-Solving Methods in Artificial Intelligence*, New York, McGraw-Hill., 1971.

Pape, U., (1974), "Implementation and Efficiency of Moore Algorithms for The shortest Route Problem," Mathematical Programming 7, 212~222.

Pearl, J., *Heuristics: Intelligent Search Strategies for Computer Problem Solving*, reading Mass: Addison Wesley, 1984.

Pohl, I., (1971), "Bi-directional Search", Machine Intelligence 6, 127~140.

Psaraftis, H. (1980), "A Dynamic Programming Solution to the Single Vehicle Many-to-many Immediate Request Dial-A-Ride Problem" Transportation Science 2, 130~154.

Rilett, L. R. (1992) *Modeling of TravTek's Dynamic Route Guidance Logic Using the Integration Model*, Ph.D. Dissertation, Queen's University, Kinston, Ontario.

Rouphail, M. N. and N. Navaneet, (1995), "Estimating Travel Time Distributions for Signalized Links: Model Development and Potential IVHS Applications" Presented at the Annual Meeting of IVHS America, Washington, D. C.

Savelsberch, W. P. M. and M. Sol, (1995), "The General Pickup and Delivery Problem" Transportation Science, Vol. 29, No. 1, 17~29.

Sedgewick, R. and J. S. Vitter, (1986), "Shortest Path In Euclidean Graphs" Algorithmic 1, 31~48.

Sexton, T. and L. Bodin (1985a), "Optimizing single vehicle many-to-many operation with desired delivery times: I. Scheduling", Transportation Science 19, 378~410, 1985a.

Sexton, T. and L. Bodin (1985b), "Optimizing single vehicle many-to-many operation with desired delivery times: I. Routing", Transportation Science 19, 378~410.

Shier, R. D., (1979), "On Algorithms for Finding the k Shortest Paths in a Network," Networks. Vol. 9, 195~214.

Sumner, R. (1991), "Data Fusion in Pathfinder and TravTek" VNIS Conference.

Tarjan, E. R., *Data Structure and Network Algorithms*. SIAM, Philadelphia, PA, 1983.

Tarko, A. and Rouphail, N. M. (1993), "Travel Time Data Fusion In Advance", ASCE Third International Conference on Applications of Advanced Technologies in transportation Engineering, Washington USA, July 1993.

Vuren, T. V. and G. R. M. Jansen, (1988), "Resent Developments In Path Finding Algorithms: A Review," Transportation Planning and Technology. Vol. 12, 57~71.

Ward, E. J. and R. E. Wendell, (1985), "Using Block Norms for Location Modeling," Operations. Research., Vol. 33, No 5.

Wilson, N. H. M. and Colvin N. H. *Computer Control of the Rochester Dial-a-ride System*. Report R77-31, Dept. of Civ. Eng. M.I.T., Cambridge, MA, 1977.

Wilson, N. H. M. and Weissberg H. *Advanced dial-a-ride algorithms research project: Final report*, Report R76-20, Dept. of Civ. Eng. M.I.T., Cambridge, MA, 1976.

CHAPTER 3

DYNAMIC AND STOCHASTIC LINK TRAVEL TIME

3.0 INTRODUCTION

For both the RGS and AVDS conceptualizations discussed in Chapter 1, one of the most important pieces of information required is the link travel times in the underlying traffic network. As described in Chapter 2, link travel times determine how the shortest path problem involved in a RGS should be defined. In an AVDS, the link travel times are the basic data used for estimating the O-D travel times which are required as basic input for the vehicle routing and scheduling problems. Due to the inherent fluctuation of travel demands, interruption of the traffic controls, unpredictable occurrences of traffic incidents and changes in weather conditions, the link travel times in an urban traffic environment may be extremely dynamic and stochastic thoroughout the day. The objective of this chapter is to examine the dynamic and stochastic attributes of the link travel times and provide some insight into how the related parameters can be obtained under various traffic conditions. The conclusions found in this chapter are used as input to Chapter 4 and Chapter 5.

As discussed in Chapter 2, the link travel time estimation and prediction problem has become one of the central focuses of various demonstration RGS projects (Hoffman and Janko, 1990; Boyce et al., 1993). However, all related research and proposed methods implicitly assume that the only relevant information on the link travel times is the average travel time for a set of discrete time intervals throughout the day and thus the link travel time variation, or the stochastic attribute is ignored.

This chapter concentrates on the dynamic and stochastic link travel time pattern on three types of links. The first type of link represents an undersaturated, uninterrupted, flow condition which prevails on most freeways and arterial sections excluding intersection interruptions (May, 1990). Section 3.1 discusses how the link travel time distribution on this type of link can be theoretically obtained from the respective speed distribution.

The second type of link, which represents a signal controlled condition, is discussed in Section 3.2. The link travel time on this type of link includes the running time on the link and the queuing delay at the intersection caused by a signal control. A simulation model is developed to examine the link travel time distribution as a function of traffic volume, signal control and platoon progression quality.

Section 3.3 discusses the third type of link which represents the traffic congestion condition caused from incidents. A stochastic model is developed to estimate the probability distribution of the incident delay, from which the mean and variance of the incident delay is derived. An example incident is created and used to analyze the performance of this new link travel time model. A sensitivity analysis of the estimation error by a deterministic model and the variance of the incident delay as a function of the incident duration is also performed. Finally, the section examines that how real time information can be incorporated in the estimation of incident delay.

It should be noted that the above three types of links by no means cover all the road facilities and traffic conditions expected in a realistic network; however, some conclusions and proposed methods may be extended to analyze the link travel time distributions on other types of links.

3.1 LINK TRAVEL TIME PATTERN UNDER UNINTERRUPTED UNDERSATURATED FLOW CONDITIONS

The link travel time during an uninterrupted, undersaturated flow condition only includes the time in which a vehicle spends in motion -- the running time. Due to certain inherent variations in traffic volumes, traffic composition and weather conditions, a vehicle may experience different link running times on a link during different times of the day (dynamic) and even at the same time over various days (stochastic). Similar to the traditional treatment of the dynamic link travel time pattern, the time horizon is divided into short intervals. The method of estimating the travel time distribution pattern in each interval is the focus of this research. Although the size of the interval is not a topic of research in this thesis, it should be noted that the interval should be wide enough to contain sufficient data for statistical inference in practical situations.

Although there is little research on the distribution of the link running times, a substantial amount of research has been conducted on the distribution of speeds— the reciprocal of the running time (TRB 1994; May, 1990). This section first compares the empirical link running time distribution to a mathematical distributions — the normal distributions. The section then discusses how running speed distributions can be used to derive running time distributions.

3.1.1 Running Time Distribution: Some Empirical Evidence

This section provides some empirical evidence regarding the vehicle running time distribution pattern on a highway section under undersaturated traffic conditions. The travel time data used in the following analysis were taken from a travel time survey conducted by the Transportation Department of the City of Edmonton in 1991. The survey data consist of link by link travel times on 9 routes during the AM peak (7:00am~9:00am), the PM peak (4:00pm~6:00pm) and the off peak periods (10:00am~12:00am) obtained during a floating car study. This research selected link travel time data on 102 Avenue from 101 Street to 100 Street which was covered by three surveyed routes with 24 observations during and AM-peak period and Off-peak time period. The frequencies of the link running time using a 4 second interval are calculated and the cumulative distribution is shown in Figure 3-1. For comparison purposes, two mathematical distributions, the normal and lognormal distributions that used the surveyed mean and variance as input parameters, are also shown in Figure 3-1. It can be observed that the link running time distribution closely resides between the normal and lognormal distributions. It should be noted that these data can be fit perfectly using some more general and powerful distributions such as a Johson Translation System. However, in order to arrive a conclusive answer on which mathematical distribution is best fit into the link running time, a intensive field survey needs to be conducted to collect enough data samples for a robust statistical analysis. Instead, for the purpose of this thesis, a theoretical analysis is performed in the following section to provide more evidence on the link running time distribution pattern.

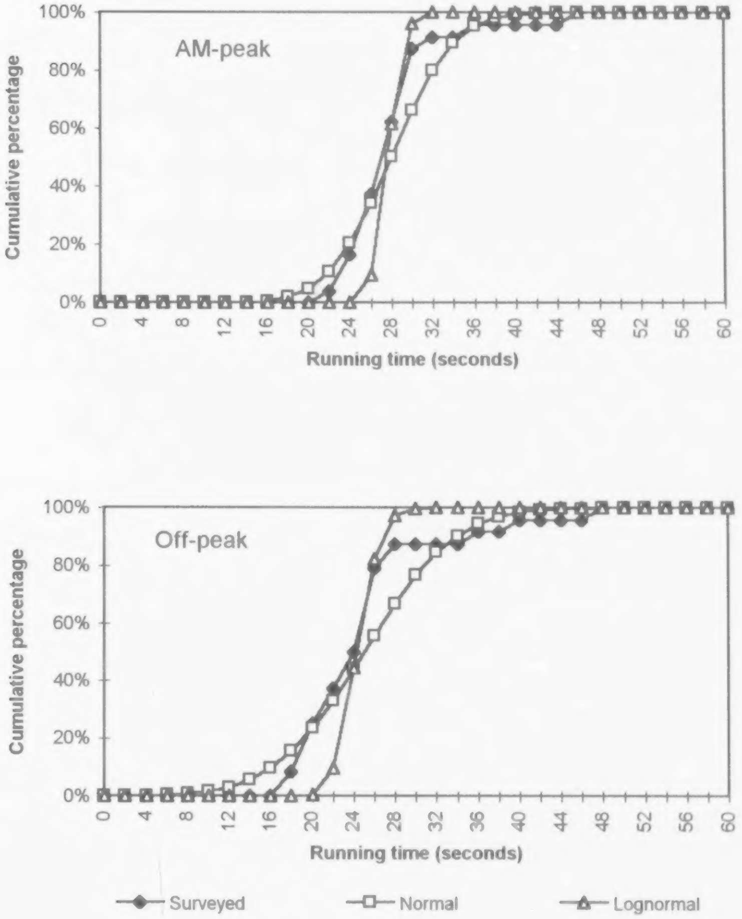


Figure 3-1 The running time distribution: surveyed vs. mathematical

3.1.2 Theoretical Derivation of the Running Time Distribution

The previous section indicates that a normal or lognormal distribution may be able to represent the vehicle running time distribution on a link. In addition, past research has suggested that the running speed, the reciprocal of the running time, is either normally distributed or lognormally distributed (Gerlough and Huber, 1975). Because running speed and running time are functionally related to each other and both are random variables it would be expected that given, the distribution of one the distribution of the other, at least theoretically, may be derived. The following paragraphs show the derivation of the running time distributions based on the running speed distributions.

1) If the running speed is normally distributed:

Consider a link with a length noted as D. The vehicle running speed on this link, S, is normally distributed with a known mean, μ_s, and a standard deviation, σ_s. The vehicle running time on the link, noted as T, is a random variable which can be calculated as follows:

$$T = \frac{D}{S} \tag{3-1}$$

In Equation (3-1) the variable D is deterministic and S is a normally distributed random variable that can be expressed as $\mu_s + \sigma_s X$, where X is a normally distributed random variable with a mean equal to zero and a standard deviation of 1, that is, X is $N\{0,1\}$. Therefore, Equation (3-1) can be rewritten as follows:

$$T = \frac{D}{\mu_s + \sigma_s X} = \frac{D/\mu_s}{1 + \nu_t X} \qquad (3\text{-}2)$$

where ν_s.is the coefficient of variation (COV) of the running speed, defined as $\nu_s = \frac{\sigma_s}{\mu_s}$.

Equation (3-2) can be expanded as a Taylor's series at the point X=0:

$$T = \frac{D}{\mu_s}\left(1 - \frac{\nu_s X}{1!} + \frac{(\nu_s X)^2}{2!} \ldots\right) \qquad (3\text{-}3)$$

If the above series is truncated at the linear term, the running time T becomes a linear function of X, and thus becomes a normally distributed variable with its mean and standard deviation defined as follows:

$$\mu_T = \frac{D}{\mu_s} \qquad (3\text{-}4)$$

$$\sigma_T = \mu_T \, \nu_s \qquad (3\text{-}5)$$

It should be noted that the quality of this approximation depends on the value of ν_s. The smaller ν_s is, the closer the running time would be to a normal distribution. A simple simulation was conducted to illustrate how close the running time is to a normal distribution under various values of the coefficient of variation (COV) for the running speed. The simulation assumes the link length is 0.5 km and average running speed is 50 km/h. Figure 3-2 shows the cumulative distributions of the simulated running time and a normal distribution based on the above approximations shown in Equation (3-4) and

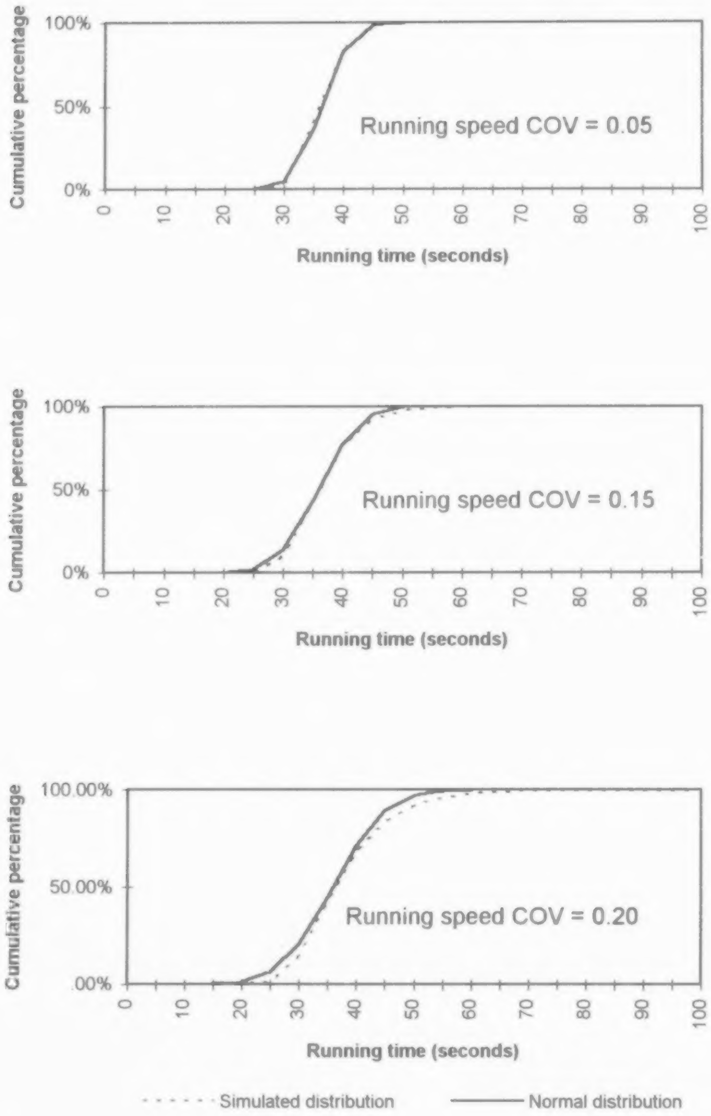


Figure 3-2 The running time distributions as compared to normal distribution

Equation (3-5) under different speed COV. It was found that when the coefficient of the running speed variation is less than 0.20, the simulated running time distributions are relatively close to the normal distribution. The chi-square test for each case, except for V, = 0.20, indicated that the running time distribution statistically shows no difference from the normal distribution at the level of significance of 0.95.

2) If the running speed is lognormally distributed:

The following equation can be obtained by a natural logarithmic operation on Equation (3-1):

$$\ln T = \ln D - \ln S \qquad (3\text{-}6)$$

From the above equation it can be found that if the running speed, S, is lognormally distributed, that is, ln S has a normal distribution, then ln T is also normally distributed or the running time is lognormally distributed. The mean and standard deviation of ln T can be obtained by the following:

$$\mu_{\ln T} = \ln D - \mu_{\ln S} \qquad (3\text{-}7)$$

$$\sigma_{\ln T} = \sigma_{\ln S} \qquad (3\text{-}8)$$

Where $\mu_{\ln S}$ and $\sigma_{\ln S}$ are respectively the mean and standard deviation of the running speed.

3.2 LINK TRAVEL TIME DISTRIBUTION UNDER A TRAFFIC SIGNAL CONTROL CONDITION

The total travel time that a vehicle spends in a road section which operates under the control of a traffic signal is comprised of two principle components: the running time and the intersection approach delay. As the running time distribution has been described in Section 3.1, this section focuses on the analysis of the distribution pattern for the intersection approach delay.

The delay in which a vehicle may experience at an intersection could have extra variation due to the signal control system and the vehicle's unpredictable arrival time at the intersection with respect to the beginning of the green interval. For example, a vehicle can go through an intersection without any delay if it arrives at the intersection during the green interval with no queue present on the approach. On the other hand, the vehicle has to wait for the entire red interval if it arrives at the beginning of the red interval.

Teply and Evans (1989) first measured the delay distribution at a signalized approach when they studied a method for evaluating signal progression quality. They found that most of the delay distributions are bimodal and therefore simple statistical parameters (e.g., mean value) cannot adequately describe these distributions. Motivated by the potential ITS applications, Rouphail and Dutt (1995) proposed a theoretical model for estimating the travel time distribution of a signalized traffic link under idealized conditions which include a constant traffic flow and a fixed traffic control. However, their models are limited with respect to the conditions they considered and in addition, they did not provide a systematic analysis of the delay distribution patterns caused from changes in traffic conditions and controls.

This thesis applied a simulation method to examine the travel time distribution patterns and their relationship with some independent factors such as traffic volume, signal timing and signal coordination. The reason is twofold. First, the traffic situation at the signalized approach involves a complicated vehicle arrival and discharge process and consequently it is impossible to develop a theoretical model to describe this situation unless a more idealized situation is assumed (Rouphail and Dutt, 1995). The second reason is that there are no signalized approach delay data available to sufficiently conduct a statistical analysis of an individual vehicle's distribution pattern under various types of traffic conditions and controls.

Similar to the running time analysis, the time horizon is divided into small intervals (e.g., five minutes), that includes at least one complete cycle. For example, if the interval is 5 minutes and the signal cycle time is 120 seconds, there is 2.5 cycles in each interval. This research and the simulation focuses on one cycle within an individual interval during which the traffic arrival or rate is relatively stable. It should be noted that the dynamic pattern is not directly discussed but reflected in the arrival rate in an interval. Furthermore, it is assumed that the exact arrival time of a vehicle within a cycle is not predictable or uniformly distributed. Figure 3-3 schematically illustrates the time horizon (time of day), time interval, signal cycle, and some hypothetical link travel times.

The following sections describe the simulation model used, verification of the simulation model and its application in a sensitivity analysis.

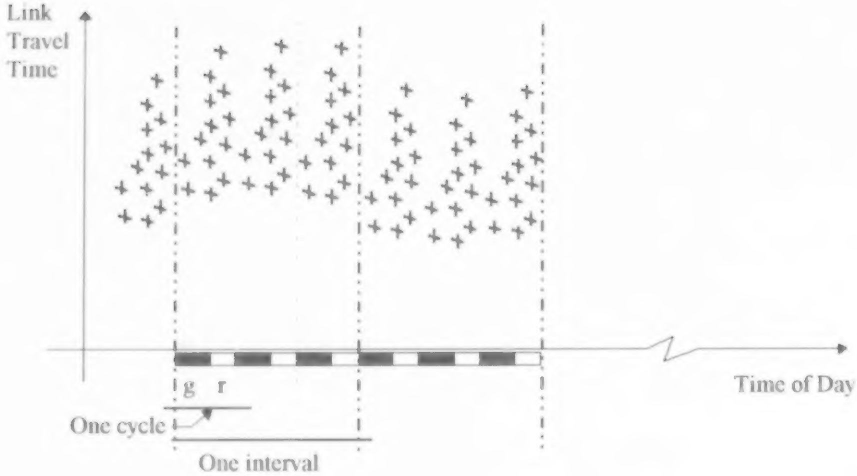


Figure 3-3 Relationship between the time horizon, time interval and cycle time

3.2.1 A Simulation Model

The simulation model explicitly models the delay that a test vehicle experiences while going through an intersection approach. The approach is assumed to be used exclusively for through traffic and controlled by a pre-timed traffic signal. The simulation focuses on the traffic operation during a time period of one cycle. The following paragraphs provide a detailed discussion of the simulation parameters and simulation procedures used.

1) The arrival pattern:

The arrival rate of the traffic is assumed to be cyclic and its cycle time is assumed to be the same as the signal cycle at the intersection. The vehicles arriving at the approach are assumed to follow two groups as shown in Figure 3-4. One group, referred to as non-platoon arrivals, includes vehicles randomly arriving at the intersection. The vehicle headway in this group can be modeled by a shifted negative distribution and the minimum headway used in this distribution model is assumed to be one second (May, 1993).

The second group consists of those vehicles arriving at the intersection in a platoon. It is assumed that these vehicles interact with each other and their headways are more likely to follow other types of distributions such as the Normal distribution or Pearson type III distribution. The initial study included three distribution cases — the constant value (deterministic), the normal distribution and the negative exponential distribution. It was found that the simulation results (i.e., delay distribution) were not sensitive to the arrival distributions chosen and therefore the constant headway assumption was used for the platoon arrivals in the following analysis.

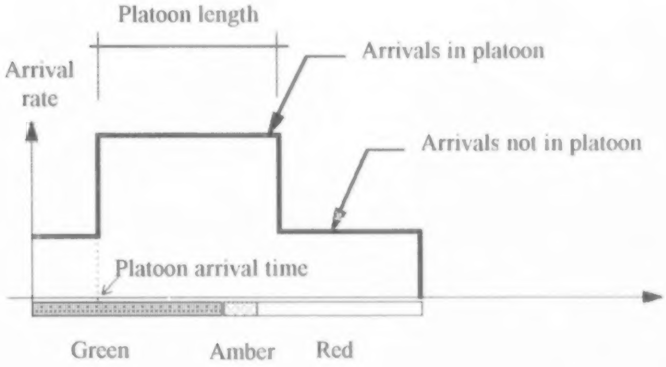


Figure 3-4 Cyclic arrival pattern at the approach

It can be seen from Figure 3-4 that two other important parameters are the platoon arrival time (time after the green starts) and the platoon size (defined as platoon duration). It can be reasonably assumed that the platoon arrival time is constant for a given approach. The platoon size, however, can vary significantly depending on such factors as the arrival pattern at the upstream intersection and the link length. This simulation uses a normal distribution to approximate the distribution of the platoon size.

2) The discharge pattern:

The vehicle discharge pattern during the green interval depends on the queue status at the approach. If there is no queue when a vehicle arrives, then it can immediately discharge with no discharge delay. Otherwise, the vehicle must wait until the vehicle ahead discharges. The discharge rate at saturation, or the saturation flow, has been found to be relatively stable (Teply *et. al.*, 1995). Therefore, a deterministic discharge headway is used in this simulation. The saturation flow used is 1800 pcu/h which corresponds to a discharge headway of two seconds.

3) Simulation procedure:

The simulation proceeds with the following logic:

- the simulation proceeds as follows. It generates each vehicle and processes it (or discharges it) before the next vehicle is generated;
- if the preceding vehicle is in a platoon and the total platoon length does not exceed a pre-generated platoon length for this arrival cycle, the new generated vehicle is considered as part of the platoon and a platoon headway is used.

Otherwise, a non-platoon headway is used. The same logic is followed when the proceeding vehicle is not in a platoon;

- if the vehicle arrives during a green interval and if there is no queue, the generated vehicle is immediately processed such that its departure time is equal to its arrival time at the intersection. Otherwise, the vehicle discharges the departure time of the vehicle ahead of it in the queue plus the discharge headway;
- at the end of an experimental run, the distribution of the individual vehicle's travel time can be obtained.

The above procedure was implemented within a Microsoft Excel spreadsheet.

3.2.2 Verification of the Simulation Model

Before the simulation program was used to analyze the link travel time distributions it is necessary to first calibrate and validate the model. The verification uses data from a traffic survey conducted by the University of Alberta (Fung, 1994). The survey examined a set of individual intersection approaches during different time periods for a time period of approximately 30 minutes. The collected data include the traffic counts of vehicles entering and exiting a specific section from an upstream reference point to the stop line on the intersection approach at 10 seconds intervals. The total travel time of an individual vehicle is calculated as the difference between the arrival time at upstream reference point and discharge time at the stop line. The individual delay is then determined by subtracting the unimpeded travel time on the section from the total travel time. Because the data are collected for 10 second intervals, the calculated delay can have as much as a 5 second estimation error. However, it can be expected that this error would not change the general distribution pattern of the delay.

Figure 3-5 shows the surveyed vehicle delay distribution as compared to the simulated results for three cases. Case a and Case b are the data from 114 Street and 76 Avenue north bound during the AM peak period and the off peak period. Case c is from 114 Street and 76 Avenue southbound during the off peak period. The parameters used in the simulation for each case are listed in Table 3-1. It should be noted that the platoon arrival time and platoon length are identified in the cumulative arrival and discharge graph for each case.

Table 3-1 Simulation parameters for model verification

	Case a	**Case b**	**Case c**
Volume (pcu/h)	944	720	520
Cycle time (seconds)	130	80	80
Green interval (seconds)	95	45	45
Saturation flow (pcu/h)	1800	1750	1600
Platoon length (seconds)	0	50	0
Platoon arrival time(seconds) (referred from start of green)	0	20	0

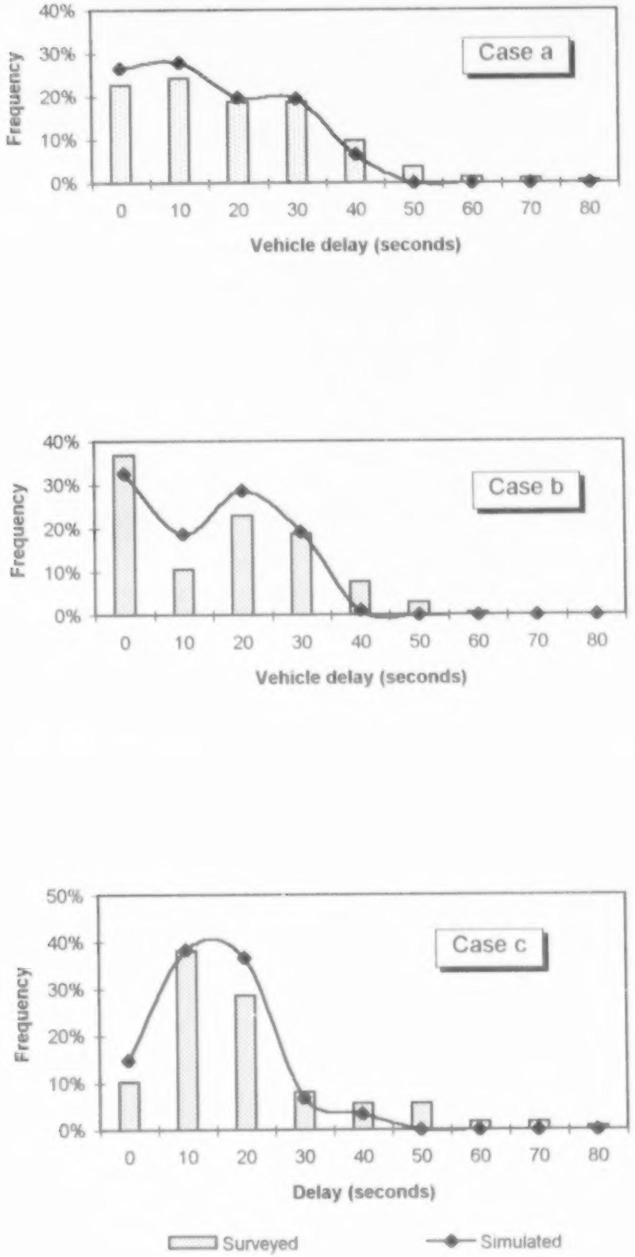


Figure 3-5 The intersection delay distributions: simulated vs. surveyed

Chi-square tests were performed on the data for the three cases. It was found that the simulated distribution and the surveyed distribution are statistically identical ($\chi^2 = 1.00 < \chi^2_{\text{table}} = 11.10$) at a confidence level of 0.95.

3.2.3 Intersection Delay Distribution: A Sensitivity Analysis

The following sections use the proposed simulation model to examine how the vehicle delay distribution pattern changes as a function of certain external factors including traffic volume, signal timing and signal control progression quality.

3.2.3.1 Vehicle delay distribution vs. traffic volume

The objective of this section is to investigate the influence of traffic volume on the distribution pattern of the individual vehicle delay at signalized intersection approaches and thus to provide insight into the delay distribution patterns under congested situations (for example, AM or PM peak period) and non-congested situations (for example, off peak period). In the simulation, it is assumed that the vehicle arrivals are purely random and that there is no cyclic platoon arrivals. The signal cycle time is set to 100 seconds with an effective green interval of 50 seconds. Figure 3-6 shows the mixed PMF and PDF that represent the probability of delay for three different traffic volumes. The discrete mass point occurs when the delay equals to zero. For example, when the traffic volume is 200 pcu/h on the link, there is an approximately 43 percent of chance that a particular vehicle will experience no delay when going through the intersection. The graph shows that the probability of experiencing no delay at the intersection significantly decreases as the traffic volume increases. The graph also reveals that the shape of the delay distribution is fairly close to a uniform distribution for the delay between zero (minimum delay) and the effective red time (or maximum delay, 50 seconds in this case) in undersaturated traffic situations.

Figure 3-7 shows the mean and standard deviation of the delay as a function of the traffic volume. As expected, the larger is the traffic volume at the approach, the larger is the average delay. However, it is interesting to see that the standard deviations are almost constant in the undersaturated situations. A possible explanation is that the variation of approach delay at an intersection is related primarily to the signal control instead of other factors such as traffic volume. This finding implies that it may not be necessary to consider the traffic volume variation in the estimation of standard deviation at a signal controlled approach.

Figure 3-8 shows the simulated vehicle delay distribution when the volume to capacity ratio is equal to 0.95 (or volume = 850pcu/h). This PMF may be compared to the normal distribution with same mean and variance. A Chi-square test was performed and it was found that the sample distribution and the normal distribution are statistically identical ($\chi^2 = 3.75 < \chi^2_{\text{table}} = 9.50$) at a 95% confidence level. This implies that when the traffic at an intersection approach is close to its capacity the vehicle delay distribution is approximately normally distributed.

3.2.3.2 Vehicle delay distribution vs. green interval

This section illustrates the relationship between the vehicle delay distribution and the effective green interval at an approach. Figure 3-9 shows the mixed PMF and PDF of the vehicle delay when the effective green interval is varied and the vehicle arrival rate is constant with an arrival rate of 500 pcu/h. As before, the signal cycle time is set to 100 seconds. It can be seen that there is a higher probability that a vehicle will not experience delay as the green interval increases. For example, when the green interval is 70 seconds for the given approach, there is approximately a 50 percent chance that a vehicle will experience zero delay when going through the intersection. As expected, the probability of experiencing zero delay at the intersection decreases as the approach is allocated less green interval. It may also be seen that the shape of the delay distribution is fairly sensitive to the green interval. As the allocated green interval decreases, the delay distribution is skewed from the left side (lower delay) to the right side (higher delay) and at the same time it becomes more probable that an individual vehicle will experience a delay longer than the effective red time (that is, a vehicle will wait more than one cycle).

Figure 3-10 shows the mean and standard deviation of the delay as a function of the effective green interval. It can be found that both the mean and standard deviation of the delay decrease as the green interval increases. However, the variation of the vehicle delay is much less sensitive to the green interval as compared to the mean of the vehicle delay. Consequently, the COV of the vehicle delay increases as the green interval increases. For example, in this case the COV is tripled when the green interval increases from 30 seconds to 70 seconds.

3.2.3.3 Vehicle delay distribution vs. the quality of the progression

The quality of the progression on the signal controlled approach is commonly recognized to have a significant impact on the average approach delay (Teply and Evans, 1989; May, 1993). This section shows the impact that the progression has on the vehicle delay distribution. As in the previous cases, a simulation study was used whereby the vehicle arrival rate is 800 pcu/h with 80 percent of the volume arriving in a platoon. The average platoon length is assumed to be 50 seconds with an associated standard deviation of 10 seconds. The variation of the platoon ratio is generated by changing the platoon arrival time at the approach. The signal cycle time is set to 100 seconds with a 50 second effective green interval.

Figure 3-11 shows the PMF and PDF under different values of the platoon ratio (Rp). It can be found that the vehicle delays with platoon arrivals are distributed more irregularly as compared to the situations without platoon arrivals. It appears that there are essentially two population groups among the vehicles. One group is composed of the vehicles with a delay less than of 10 seconds and the vehicles in the other group have a delay of approximately 50 seconds. In the case of good coordination (higher Rp) most of the vehicles have significantly lower delay as compared to the situation of low platoon ratio. In this latter case the majority of the vehicles arrive during the red phase and consequently are delayed for longer periods.

Figure 3-12 illustrates the relationship between the mean and standard deviation of the delay and the progression ratio. It can be seen that the average delay decreases approximately at a constant rate as the progression ratio increases. This result is consistent with the consideration of the platoon ratio included in delay calculation in Highway Capacity Manual (TRB, 1994). Figure 3-12 also indicates that both good coordination and bad coordination will result in a slightly lower standard deviation (less than five seconds). This implies that the signal coordination may be disregarded in the estimation of standard deviation.

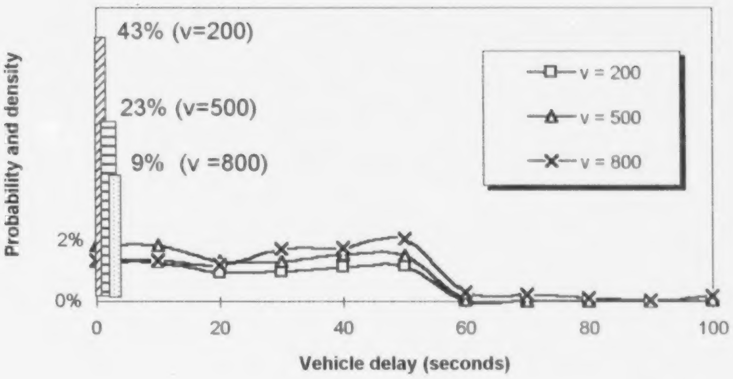


Figure 3-6 The mixed PMF and PDF of the vehicle delay under different traffic volumes

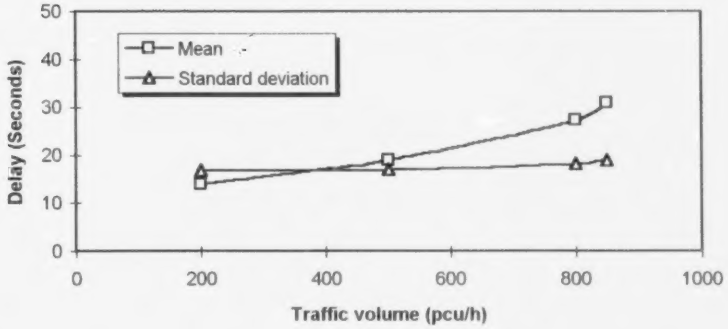


Figure 3-7 The relationship between the mean and standard deviation of the vehicle delay and traffic volume

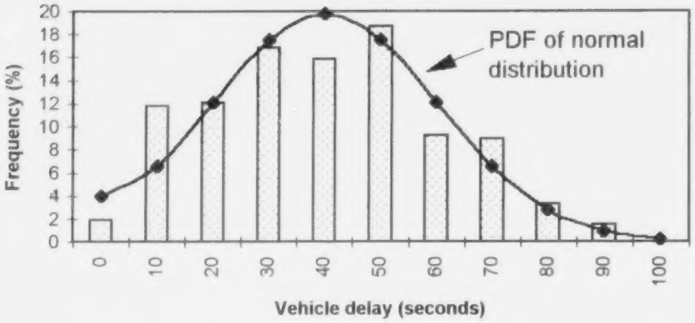


Figure 3-8 Vehicle delay distribution as compared to normal distribution when v/c=0.95

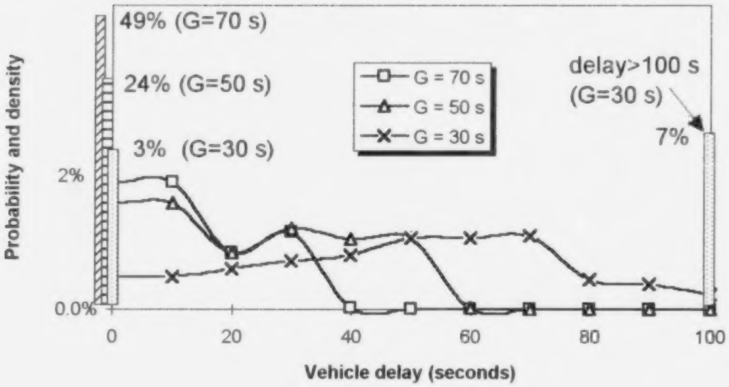


Figure 3-9 The mixed PMF and PDF of the vehicle delay under different green time

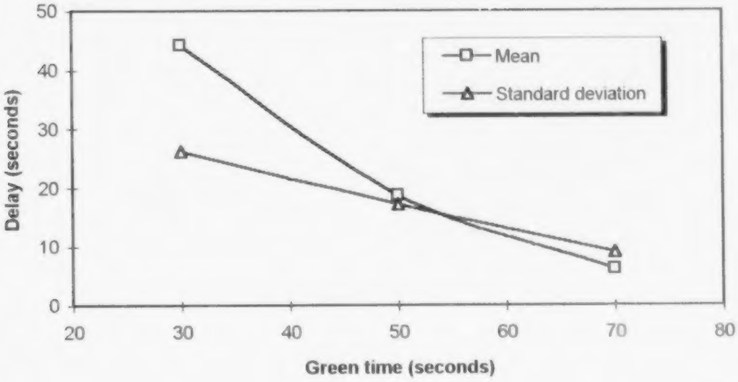


Figure 3-10 Relationship between the mean and standard deviation of the delay and green time

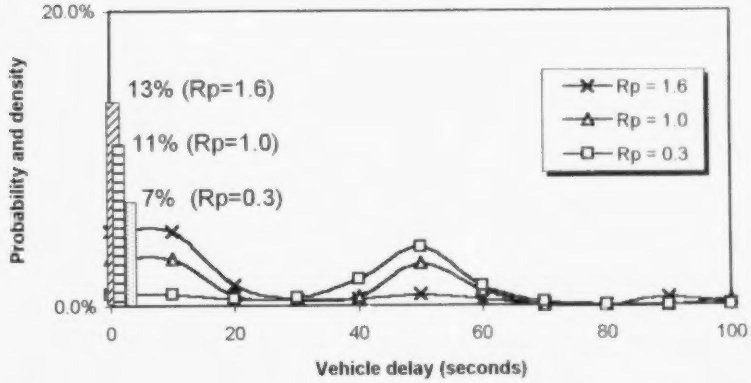


Figure 3-11 The mixed PMF and PDF of the vehicle delay under different quality of progression

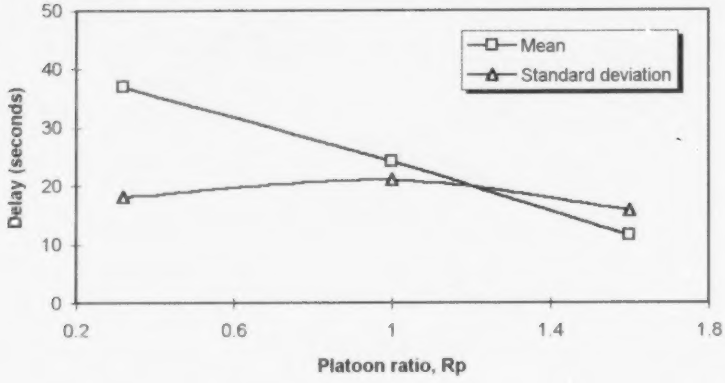


Figure 3-12 Relationship between the mean and standard deviation of the delay and the platoon ratio

3.3 LINK TRAVEL TIME DISTRIBUTION UNDER INCIDENT CONGESTION

The previous sections have discussed the distribution patterns of the link running time and intersection delay that a vehicle may experience under a normal traffic condition. Another major delay a vehicle may encounter is the incident delay resulting from incidents. Because traffic incidents are essentially rare and random events, they need to be considered in real time vehicle routing procedures only when they are realized (or detected). Accordingly, this research considers the situation that an incident has been detected and tries to estimate the distribution pattern of the incident delay that a vehicle may experience when it arrives at the incident spot.

Traditionally, incident delay is estimated by using a deterministic queuing model which assumes that the traffic situation (i.e., the arrival rate) and incident situation (i.e., the reduced capacity and incident duration) can be exactly identified (Al-Deek and Kanafani,1991; Koutsopoulos and Yabouski, 1991). This approach may be adequate for an "after" evaluation where the information on the traffic volume and incident situation are available. However, it is inappropriate for the prediction of incident delay in real-time applications such as a dynamic RGS. In this later situation, the only information available for link travel time prediction might consist of the time when an incident occurs (detected), the current situation of the incident (removed or not and reduced capacity) and traffic volume. Obviously, the length of the incident is unknown. Therefore, the incident delay is a dynamic and stochastic variable, and might not be correctly estimated by a deterministic model. Another potential drawback to the deterministic model is that the model does not consider the variance of incident delay which is significant under the incident congestion and clearly important for vehicle routing decisions.

This section developed an incident delay estimation model which explicitly considers the randomness of the incident duration. The ultimate goal will be to use this model to estimate the dynamic and stochastic incident delay (e.g., represented as mean and variance) which can be used in the new vehicle routing models discussed in Chapter 4 and Chapter 5.

3.3.1 Assumptions and Notation

The incident delay experienced by an individual vehicle that goes through an incident location depends on many factors. The main factors include incident severity (capacity reduction), incident duration, the traffic volume and the time when the vehicle arrives at the incident location. All of these factors may be considered as random variables in a real-time situation, which makes the estimation procedure of the incident delay very complicated. To simplify the model development, the following assumptions are used:

1) The traffic arrival rate at the incident location is constant and can be estimated exactly;
2) The capacity reduction caused by the incident is constant and can be detected exactly;
3) The link is long enough so that there will be no spill back to the upstream link;
4) The incident duration is a random variable and its probability distribution function (PDF) can be identified from a historical incident database;
5) If the incident is over and the capacity is restored to its original value, then the prediction problem becomes a simple deterministic problem, and hence is not considered in this derivation.

Based on above assumptions, a stochastic model was developed by treating incident duration as a random variable within a typical deterministic queuing model. Figure 3-13 is a queueing diagram showing the cumulative vehicle arrivals and vehicle departures before and after the incident is cleared. It also illustrates some parameters used in the following analysis. The parameters used are defined as follows:

Basic parameters: The values of these parameters are known and used as basic input:

q = average link flow over the time period when the incident impact prevails (pcu/h);

c = link capacity under non-incident conditions (pcu/h);

c^* = reduced link capacity caused by an incident (pcu/h);

T^* = time when an incident occurs ;

T_0 = current time (or time when the prediction is required);

D^* = incident duration, a random variable with known PDF. Based on assumption 5, $D^* \geq T_0 - T^*$;

$f_{D^*}(x)$ = PDF of the incident duration (D^*);

T_a = estimated time when an vehicle arrives at the incident spot on the link;

Derived parameters: The value of these parameters can be calculated with input of the basic parameters (refer to Figure 3-14):

T_1 = a random variable representing the time point at which a maximum delay occurs. It can be represented as a function of the random incident duration D^*:

$$T_1 = T^* + D^* c^*/q \tag{3-9}$$

T_2 = a random variable representing the time when the incident is cleared, it can also be represented as a function of the incident duration D^*:

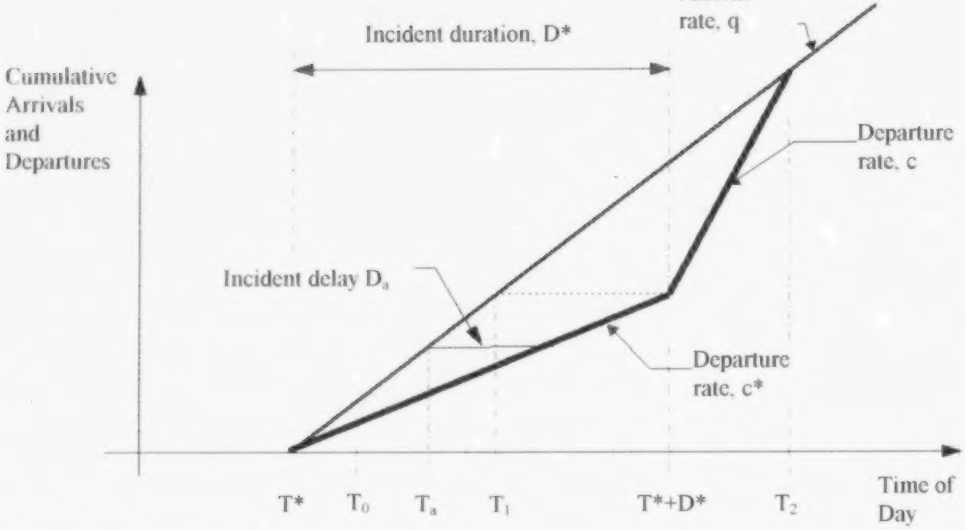

Arrival rate, q
Incident duration, D*
Cumulative Arrivals and Departures
Departure rate, c
Incident delay D_a
Departure rate, c*
T*
T_0
T_a
T_1
T*+D*
T_2
Time of Day

Figure 3-13 Queuing model of the incident delay and the related parameters

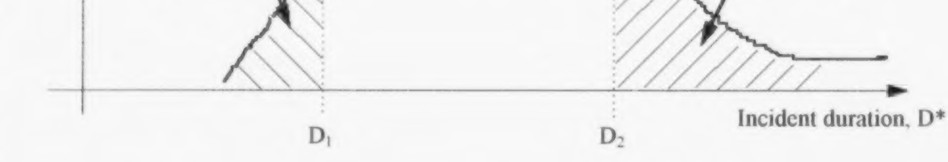


Figure 3-14 The density function of the incident duration and the parameters related to incident delay

$$T_2 = T^* + \left(\frac{c - c^*}{c - q}\right) D^* \qquad (3\text{-}10)$$

D_1 = incident duration which makes the arrival time of an individual vehicle coincident with the time when the incident is cleared, i.e. $T_a = T_2$. It can be derived from Equation (3-10):

$$D_1 = \frac{c - q}{c - c^*} (T_a - T^*) \qquad (3\text{-}11)$$

D_2 = incident duration which makes the arrival time of an individual vehicle (T_a) coincident with the time when a maximum delay occurs, i.e., $T_a = T_1$. It can be derived from Equation (3-9):

$$D_2 = \frac{q}{c^*} (T_a - T^*) \qquad (3\text{-}12)$$

P_1 = probability that the incident duration is less than D_1 :

$$P_1 = P(D^* < D_1) = \int_0^{D_1} f_{D^*}(x)\,dx \qquad (3\text{-}13)$$

P_2 = probability that the incident duration is greater than D_2 :

$$P_2 = P(D^* > D_2) = \int_{D_2}^{\infty} f_{D^*}(x)\,dx \qquad (3\text{-}14)$$

P_3 = probability that the incident duration is greater than D_1 and less than D_2:

$$P_3 = 1 - P_1 - P_2 \qquad (3\text{-}15)$$

d_m = maximum possible incident delay when a vehicle arrives at the incident spot at time T_a, it occurs when incident duration is greater than D_2:

$$d_m = \frac{q - c^*}{c^*} \cdot (T_a - T^*) \tag{3-16}$$

D_{12} = conditional expectation of the incident duration, which is defined as:

$$D_{12} = \int_{D_1}^{D_2} x f_{D^*}(x) dx \tag{3-17}$$

V_{12} = conditional expectation of the squared incident duration, which is defined as:

$$V_{12} = \int_{D_1}^{D_2} x^2 f_{D^*}(x) dx \tag{3-18}$$

D_a = incident delay experienced by a vehicle which arrives at the incident spot at time T_a, it is a random variable and its distribution needs to be found

3.3.2 Probability Distribution of Incident Delay

The probability distribution of the incident delay (D_a) depends on the probability distribution pattern of the incident duration. Their relationship can be established by analyzing how the incident delay is calculated. As shown in the queuing diagram in Figure 3-13, the incident delay can be grouped into three situations:

(1). If a vehicle arrives at the time that the incident and the resulting queue has been cleared, i.e. $T_a \geq T_2$, it will experience no delay. Therefore the probability that the delay equals zero is the same as the probability of $T_a \geq T_2$, or,

$$P(D_a = 0) = P(T_2 \geq T_a)$$

substitute for T_2 using Equation (3-10) gives:

$$\begin{aligned} P(D_a = 0) &= P(D^* \leq \frac{c-q}{c-c^*}(T_a - T^*)) \\ &= P(D^* \leq D_1) \\ &= P_1 \end{aligned} \qquad (3\text{-}19)$$

Where P_1 is defined in Equation (3-13).

(2) If a vehicle arrives at time T_a in the range of $T_0 \leq T_a \leq T_1$, it will experience a fixed amount of delay (d_m) that can be calculated using Equation (3-16). Therefore:

$$D_a = d_m = \frac{q-c^*}{c^*} \cdot (T_a - T^*)$$

the probability of delay at this point is equal to the probability that $T_0 \leq T_a \leq T_1$, or,

$$P(D_a = d_m) = P(T_0 \leq T_a \leq T_1) = P(T_a \leq T_1)$$

substituting for T_1 using Equation (3-9) gives:

$$\begin{aligned} P(D_a = d_m) &= P(D^* \geq \frac{q}{c^*}(T_a - T^*)) \\ &= P(D^* \geq D2) \\ &= P_2 \end{aligned} \qquad (3\text{-}20)$$

Where P_2 is defined in Equation (3-14).

(3) If a vehicle arrives at time T_a in the range of $T_1 < T_a < T_2$, it will experience a variable delay depending on the incident duration, which is expressed as:

$$D_a = \frac{c-c^*}{c}\cdot D^* - \frac{c-q}{c}\cdot(T_a - T^*) \tag{3-21}$$

Based on Equation (3-9) and Equation (3-10), the condition $T_1 < T_a < T_2$ can be transformed into:

$$\frac{c-q}{c-c^*}(T_a - T^*) < D^* < \frac{q}{c^*}(T_a - T^*)$$

or, $D_1 < D^* < D_2$ based on the definitions shown in equations (3-11) and 3-12.

Correspondingly, the incident delay (D_a) has the range:

$$0 < D_a < d_m \tag{3-22}$$

Because the PDF of incident duration (D^*) is known, the PDF of the incident delay (D_a) can be derived from Equation (3-21):

$$f_{Da}(x) = \frac{c}{c-c^*}\cdot f_{D^*}\left(\frac{c}{c-c^*}x + D_1\right) \tag{3-23}$$

In conclusion, the incident delay (D_a) is a mixed discrete and continuous random variable with a distribution function decided by a probability function (Equation (3-19) and Equation (3-20)) and a density function (Equation(3-23)). Figure 3-15 schematically illustrates the distribution pattern of the incident delay.

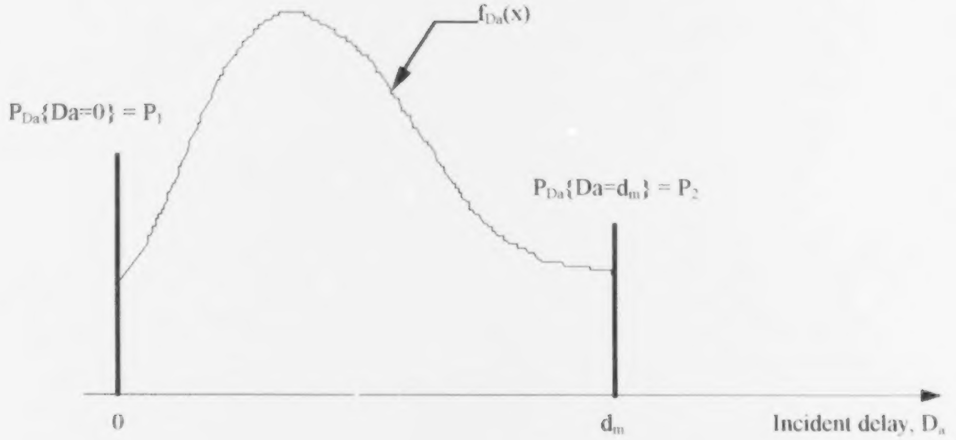


Figure 3-15 The mixed PMF and PDF of the incident delay

3.3.3 Mean and Variance of Incident Delay

In the above section the probability distribution of the incident delay has been derived. However, in the majority of real applications, the more reasonable information about the random incident delay is its first and second moments, i.e. the mean and variance of the incident delay. With the given distribution functions, these two descriptors can be obtained through the following mathematical expectations:

$$E[D_a] = 0 \cdot P(D_a = 0) + d_m \cdot P(D_a = d_m) + \int_0^{d_m} f_{D_a}(x)\, x\, dx \tag{3-24}$$

$$VAR[D_a] = E[D_a^2] - (E[D_a])^2 \tag{3-25}$$

where,

$$E[D_a^2] = 0^2 \cdot P(D_a = 0) + d_a^2 \cdot P(D_a = d_m) + \int_0^{d_m} f_{D_a}(x) x^2 \, dx$$

using equations (3-9) (3-10) (3-13) gives:

$$E[D_a] = P_2 \, d_m + \frac{c - c^*}{c} (D_{12} - D_1 \, P_3) \tag{3-26}$$

$$E[D_a^2] = P_2 \, d_m^2 + (\frac{c - c^*}{c})^2 (V_{12} + D_1^2 \, P_3 - 2 \, D_1 \, D_{12}) \tag{3-27}$$

where all the parameters are defined in section 3.3.1.

3.3.4 Expected Incident Delay: A Comparison to the Deterministic Incident Delay Model

Incident delay is traditionally estimated by using a deterministic model which assumes that the attributes of an incident (capacity reduction, duration) are known or can be estimated exactly. A deterministic queuing model is used to estimate the amount of delay. If the average incident duration used is μ*, the incident delay (D$_a$) is calculated by the following formula (Figure 3-13):

$$D_a = \begin{cases} 0 & \text{if } T_a \geq T_2 \\ \dfrac{q - c^*}{c^*} (T_a - T^*) & \text{if } T_a < T_2 \\ \dfrac{c - c^*}{c} \mu^* - \dfrac{c - q}{c} (T_a - T^*) & \text{if } T_1 < T_a < T \end{cases} \tag{3-28}$$

where all the other symbols are defined in section 3.3.1 with D* replaced by its mean, μ*.

Apart from the fact that the deterministic model does not provide information on the incident variation, it will also generate a biased estimation on the mean incident delay as show in Equation (3-24) and Equation (3-28). A numerical example is created to demonstrate the estimation biases of the deterministic model and its sensitivity to the variation of the incident duration.

Assume that there is a one-way two lane highway with capacity equal to 3600 pcu/h. An accident was just detected on the road, which reduced the highway capacity to 1800 pcu/h. The average traffic volume under normal traffic flow conditions during this time period is estimated to be 3000 pcu/h among which approximately 500 pch/h are assumed to diverts to other routes due to the incident. Furthermore, it is be assumed that the incident duration is lognormally distributed with a mean incident duration of 30 minutes and a standard deviation ranging from 0 to 30 minutes.

The incident delay, calculated using Equation (3-24), as a function of the variance is illustrated in Figure 3-16. From Figure 3-16, it can be seen that, as expected, the mean delays estimated by both models are exactly the same when there is no variation in the incident duration. However, as the variation of the incident duration increases, the result from the stochastic model is significantly different from that of the deterministic model. The deterministic model may over-estimate and under-estimate the incident delay depending on the arrival time. For example, if the standard deviation of the incident duration is 20 minutes (Figure 3-16(c)), the deterministic model would over-estimate the expected incident delay by as much as 50 percent for a trip arriving at the incident spot 20 minutes after the incident occurs.

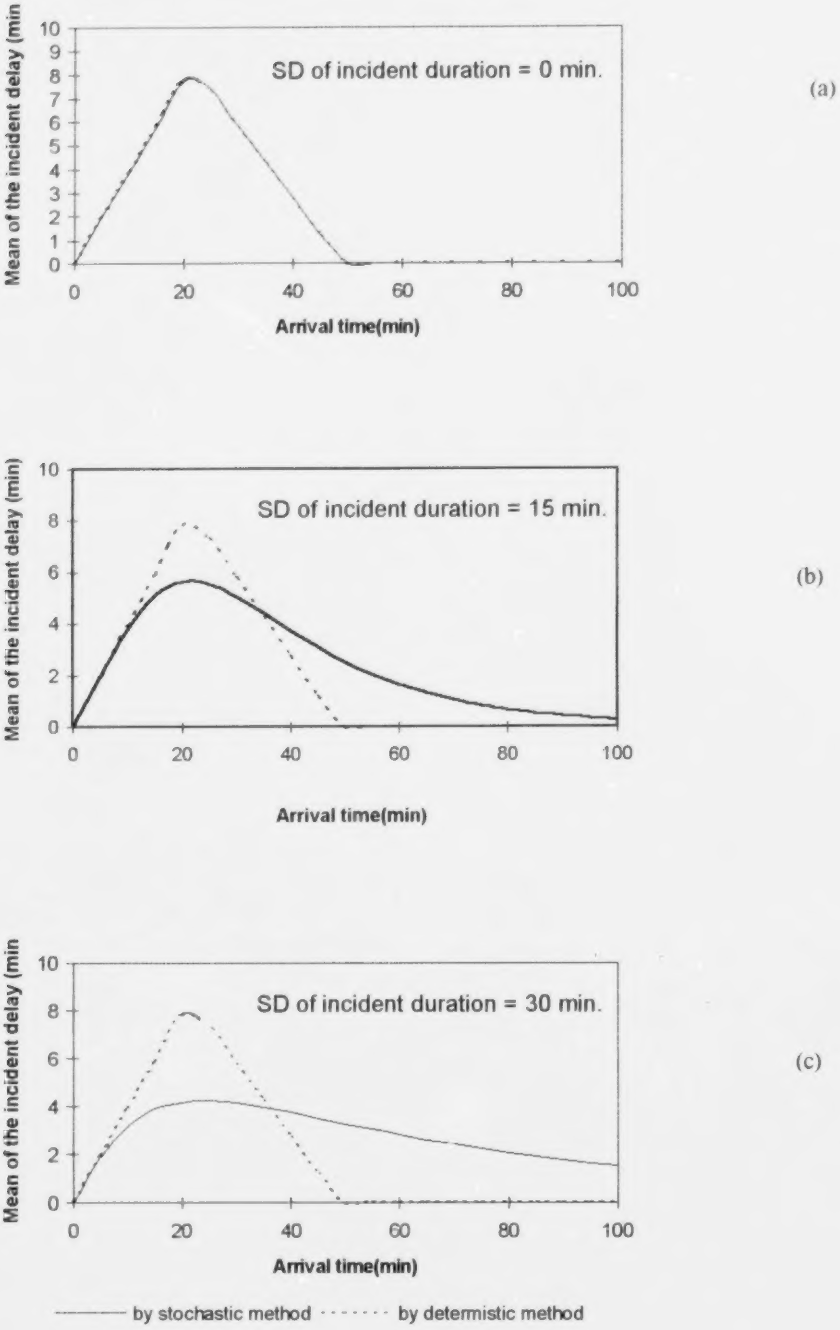


Figure 3-16 Estimation of the expected incident delay: deterministic model vs. stochastic model

If instead the trip arrives at the incident spot after 40 minutes, the deterministic model would under-estimate the expected incident delay by over 50 percent. Figure 3-16 shows the relationship between the maximum over-estimation error and under-estimation error with the standard deviation of the incident duration. The estimation error is defined as the ratio of the difference in the expected delays estimated by the deterministic and stochastic models to the expected delay by the stochastic model. As shown in Figure 3-17, the estimation error is virtually proportional to the variation of the incident duration.

3.3.5 Variation of Incident Delay

Similar to the expected length of incident delay, the variation of incident delay is also dependent on the variation of the incident duration. Figure 3-18 illustrates the standard deviation of the incident delay as a function of the arrival time at the incident location under different variances of the incident duration. The data were obtained from the same example as presented in the previous sections. As expected, the larger the variation of the incident duration, the larger the variance of the incident delay.

From Figure 3-18 it can be seen that there is a large amount of variation in the average incident delay and that this variation is more significant for the trips arriving after the expected incident duration time. For example, when the standard deviation of the incident duration is greater than 15 minutes, the COV of the incident delay for trips arriving after the expected incident duration (30 minutes) have values larger than 2.0.

Another fact that can be observed is that although the expected incident delay is small when a vehicle arrives around the expected incident clearance period, the variation of the incident delay could be very large. As shown in Figure 3-18(c), when the arrival time is 80 minutes after the occurrence of the incident, the expected delay is as small as 2 minutes, however the standard deviation reaches 6 minutes. One indication of this fact is that a routing decision based on the average travel time would provide a route with a higher order of risk resulting in an inferior decision.

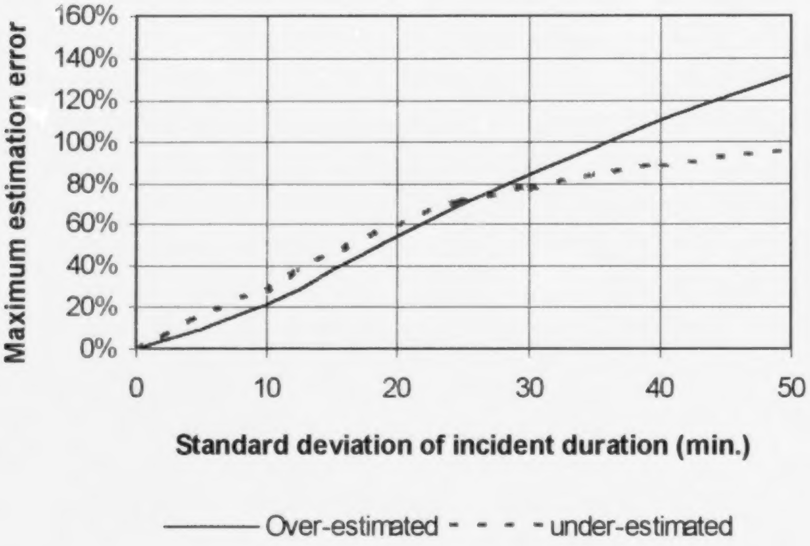


Figure 3-17 Estimation error of the incident delay by a deterministic model

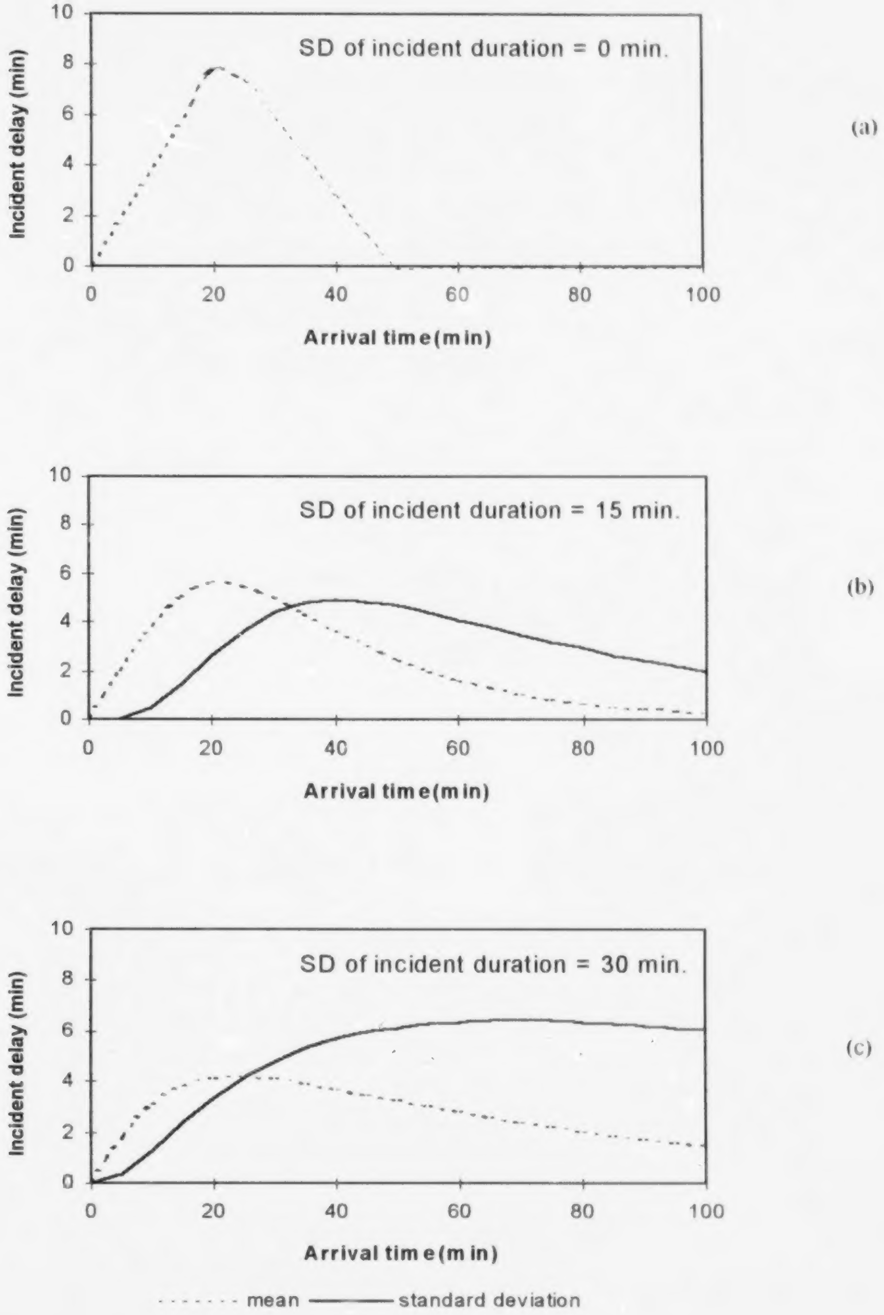


Figure 3-18 Estimation of the incident delay: mean and standard deviation

3.3.6 Incident Duration: Prior and Posterior Probability Distribution

As discussed in previous sections, one of the major pieces of information required for estimating the incident delay is the incident duration distribution. It can be expected that the incident duration can have a very high variation depending on the incident managing capability of the local authority, the incident location and the incident severity among other factors. However, it is still feasible to establish location specific distribution functions based on historical data (Golob et. al., 1987; Giuliano, 1989). Conversely, the information on the incident status (i.e., removed or not?) may also be available in most cases. Under the context of ITS, such information may be managed by a traffic information center (TIC) as discussed in Chapter 1. The following sections focus on how the ability to update information can be used to improve the estimation of the probability distribution of incident duration and how it may be applied in the incident delay estimation model developed above.

(1) Prior probability distribution of incident duration:

Some previous theoretical and empirical work (Golob et. al., 1987; Giuliano, 1989) have shown that the incident duration typically has a lognormal distribution. This research therefore assumes that the incident duration is lognormally distributed and its distribution can be established and categorized if necessary. These incident duration distributions can be considered as prior knowledge on the incident duration. If the mean of the natural logarithmic of the incident duration ($\ln(D^*)$) is λ and the standard deviation of $\ln(D^*)$ is ξ, then $\ln(D^*)$ is $N\{\lambda,\xi\}$ with density function noted as $f'_{D^*}(x)$.

(2) Posterior probability distribution of incident duration:

Assume that the TIC at the current time (T_0) receives new information showing that an incident still has not been cleared since its occurrence at time T*. The implication of this information is that the incident duration must be longer than (T_0-T*). Therefore, the probability distribution of the incident duration should be modified to take into account this new information. The modified PDF of the incident duration, i.e. posterior PDF ($f'_{D*}(x)$), can be obtained by applying Bayesian theory:

$$f''_{D*}(x) = k \cdot L(x) \cdot f'_{D*}(x) \qquad (3\text{-}29)$$

where:

L(x) = the likelihood function of the observed output, which is:

$$L(x) = \begin{cases} 0 & if\ x \le T_0 \\ 1 & if\ x > T_0 \end{cases}$$

k = a constant defined as follows:

$$k = \left[\int_{-\infty}^{\infty} L(x) f'_{D*}(x)\,dx \right]^{-1} = \left[\int_{T_0}^{\infty} f'_{D*}(x)\,dx \right]^{-1}$$

$$= \left[1 - \phi\left(\frac{\ln(T_0) - \lambda}{\zeta} \right) \right]^{-1}$$

Figure 3-19 schematically illustrates the relationship between the prior PDF, posterior PDF and the likelihood function. It should be noted that the above method can also be extended to incorporate other types of information on the incident situation such as an estimation of incident duration from an emergency vehicle operator or police officer.

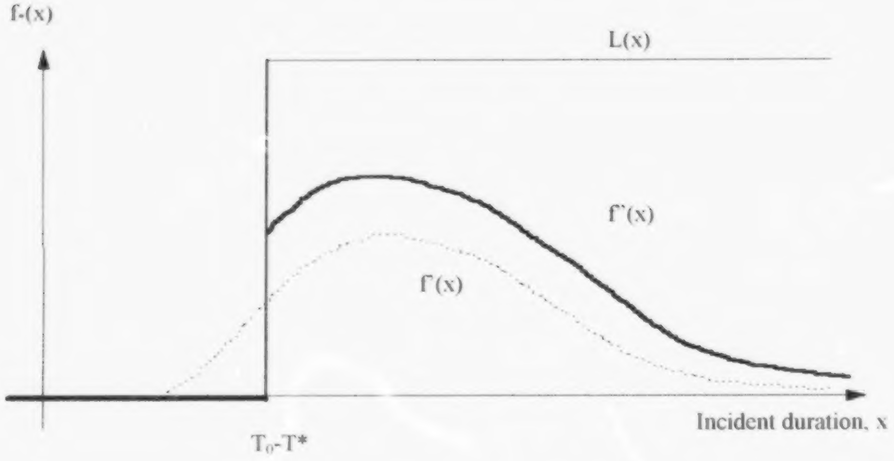


Figure 3-19 Prior and posterior distribution function of the incident duration

3.4 CONCLUSIONS

This chapter discussed the dynamic and stochastic link travel time patterns of three special cases. The conclusions are summarized as follows.

1. On the link running time distribution:
   - This research showed both theoretically and empirically that the link running time can be represented by a normal or lognormal distribution;
   - The link running time approximately has the same type of distribution as the link running speed. Consequently, the distribution parameters of the link running time can be indirectly obtained using data on the link running speed. This finding could be important when the link running speed is easier to obtain.

2. On the distribution of the signalized approach delay:

   - A simulation model has been developed to analyze the distribution pattern of the delay that a vehicle may experience at a signalized intersection;
   - It was found that the signalized approach delay is a mixed random variable and its distribution pattern can not be approximated by a single mathematical distribution or distribution family. When vehicles arrive cyclically at the approach and some of the vehicles arrives in platoon, the delay distribution may be bimodal. This latter effect is particularly noticeable when coordination is poor;
   - The variance of the vehicle delay has been found to be insensitive to the traffic volume and the quality of progression, but it is affected by the signal setting. Therefore, for an intersection that has a fixed signal timing, a single value of vehicle delay variance may be used;
   - When the traffic at the signalized approach is close to a saturated condition, the vehicle delay can be approximated by a normal distribution.

3. On the incident delay distribution:

   - This chapter developed a stochastic model for the prediction of the congestion delay caused by an incident. In contrast to the historical deterministic model, the new model explicitly considered the stochastic attribute of the incident duration. The derived formula for calculating the mean and variance of the incident delay requires only a minor amount of additional data and computational effort, and therefore may be used in some real applications;

- A deterministic model may over-estimate or under-estimate the expected incident delay, depending on the arrival time. The maximum estimation error is proportional to the standard deviation of the incident duration;
- The incident delay has been shown a high degree of variability, even when the expected delay is low. The maximum variance occurs much later than when the time of the maximum expected delay occurs;
- The new model can also use the updated information on the incident situation. This is done by modifying the PDF of the incident duration based on Bayesian theory;

REFERENCES:

Al-Deek H. and Kanafani A., (1991), "Incident Management with Advanced Traveler Information System", *Proceedings of the Second Vehicle Navigation and Information System Conference*, VINS'91, Dearborn, MI.

Boyce, D., Rouphail, N., and A. Kirson, (1993), "Estimation and Measurement of Link Travel Times in the ADVANCE Project", *IEEE-IEE Vehicle Navigation & Information Systems Conference*, Ottawa - VINS'93.

Fung, J. (1994) "Traffic Queues at Signalized Intersections", Course report, Department of Civil Engineering, University of Alberta, Edmonton, Alberta.

Gerlough, L. D. and M. J. Huber, *Traffic Flow Theory*, Transportation Research Board, Special Report 163, TRB, Washington, D. C., 1975.

Giuliano G., (1989), "Incident Characteristics, Frequency, and Duration a High Volume Urban Freeway", *Transportation Research -A*, **23A**, No. 5, 387~396.

Golob T. F., Recker W. W. and Leonard J. D. (1987), "An analysis of the severity and incident duration of truck-involved freeway accidents" *Accident Analysis and Prediction* **19**, 375~395.

Hoffman C. and Janko J. (1990), "Travel Time as a Basic of the LISB Guidance Strategy" Paper presented at the IEEE road Traffic Control Conference. London.

Koutsopoulos H. N. and Yablonski A. (1991), "Design Parameters of Advanced Information System: the Case of Incident Congestion and Small Market Penetration", *Proceedings of the Second Vehicle Navigation and Information System Conference*, VINS'91, Dearborn, MI.

Koutsopoulos H. N. and Xu H. (1991), "An Information Discounting Routing Strategy for Advanced Traveler Information Systems", *Transportation Research -C*, No. **3**, 249~264.

May, D. A. *Traffic Flow Fundamentals* Prentice Hall, Englewood Cliffs, New Jersey, 1990.

Rouphail, M. N and N. Dutt, (1995), "Estimating Travel Time Distribution for Signalized Links: Model Development and Potential IVHS Applications", Presented at the Annual Meeting of IVHS America, Washington, D. C., March 15-17, 1995.

TRB, (1994), *Highway Capacity Manual*, 1994 Update, Special Report 209, Transportation Research Board, National Research Council, Washington, D. C., 1985.

Teply, S., D. I. Allingham, D. B. Richardson and B. W. Stephenson, *Canadian Capacity Guide for Signalized Intersections*, Second Edition (S. Teply, ed.), Institute of Transportation Engineering, District 7, Canada, 1995.

Teply, S. and G. D. Evans, (1989), "Evaluation of the Quality of Signal Progression by Delay Distributions." *Transportation Research Record* **1225**, TRB, Washington D. C., 1989.

CHAPTER 4

ESTIMATION OF ROUTE TRAVEL TIME IN DYNAMIC AND STOCHASTIC TRAFFIC NETWORKS

4.0 INTRODUCTION

Central to any RGS are the shortest path algorithms required to calculate the optimal route from an origin node to a destination node given the underlying traffic network data. The development of the appropriate shortest path algorithms is directly related to how the travel time of a given route in the network can be calculated, or the relationship between the route travel time and the link travel times. For example, it has been found that standard shortest path algorithms can be equally applied to find the shortest paths in a dynamic network (Drefus,1969 ; Kauman and Smith ,1990). This finding is essentially based on the addition property of the route travel time in a dynamic network. That is, the travel time of a route is the summation of the link travel times along the route. It is because of this addition property that makes Bellman's "principle of optimality" hold. Therefore, it is necessary to examine the relationship between the route travel time and the link travel times in a dynamic and stochastic traffic network before algorithms can be developed for solving the shortest path problem in this type of network. The models developed in this chapter will be used as the foundation of Chapter 5 to develop solution methods to the shortest path problem in a dynamic and stochastic network.

The calculation of the route travel time based on the link travel times is also important for solving the DARP when the dynamic and stochastic attribute of the O-D travel time is explicitly modeled. The objective of this chapter is to investigate this relationship in a network where the link travel times are both dynamic and stochastic, and more specifically, to develop methods for estimating the mean and variance of the travel time of a given route based on link travel time data.

The problem of calculating the travel time of a given route in a traffic network has not been directly investigated because it is trivial when the link travel times in the network are assumed to be deterministic or independent random variables. Under such assumptions, for example, the mean and variance of route travel time are simply the summation of the mean and variance of link travel times along the route (i.e., the addition property holds). However, when the traffic network is modeled as both dynamic and stochastic, such addition property of the route travel time may no longer exist due to the non-linear nature of the link travel time.

This chapter first defines the context of a dynamic and stochastic network and the route travel time estimation problem in this type of network. A complete notation and some important assumptions are described. Next, this chapter derives a probabilistically based method for estimating the mean and variance of route travel time using the mean and variance of the individual link travel times. The method explicitly accounts for the dynamic nature of the link travel times and any correlation between links on the route.

Approximation models are proposed for use in real applications and the issues related to the practical application of these models are also addressed.

Lastly, the approximation models are compared with the traditional methods, both theoretically and empirically. The empirical analysis is based on a simulation study using the Edmonton network as the test bed.

4.1 PROBLEM DEFINITION

Consider a road network composed of nodes and links. Each link in the network has an associated generalized cost. In this thesis the travel time will be used to represent this generalized cost. The dynamic and stochastic attributes of a traffic network are defined by modeling the link travel time on some or all of the links in the network as a stochastic process, noted as $\{t_T , T\}$. That is, the link travel time probability distributions are dependent on the arrival time (or time of day, T) at the starting node of a link. Furthermore, this thesis assumes that the link travel times are continuous random variables and the only available information about their distribution is their respective means (noted as $\mu_{(T)} = E[t_T|T]$) and variances (noted as $\sigma^2_{(T)} = VAR[t_T|T]$). Therefore, the link travel times will be described solely by their time-dependent means and variances. It should be noted that the subscript that specifies the link is deliberately omitted for notation convenience. The methods available to estimate the mean and variance of the travel time of a given route with a given departure time in this type network is the focus of this chapter.

Because this thesis considers the travel time estimation problem of a given route, a path starting from an origin node 1 to a destination node N is used to define the problem. As shown in Figure 4-1 the path goes though node i and node i+1 by link (i,i+1). Assume that a series of travel experiments are conducted along this path. Each experiment represents a procedure of departing from the origin node 1 at the exact same time (T_1) and traveling along the path to the destination node N. Due to the dynamic and stochastic attributes, the outcomes of the experiment such as arrival time at each node and travel time on each link are random variables. These random variables are defined as follows:

T_i = a random variable indicating the arrival time at node i. Assume that there is no waiting time at the node, then T_i equals to the time entering the link (i,i+1), or departure time at node i. In addition, the departure at the origin node 1, T_1, is assumed to be deterministic and known a priori;

$f(T_i)$ = the probability density function of T_i;

$E[T_i]$ = the expected arrival time or departure time at node i ;

$Var[T_i]$ = the variance of the arrival time or departure time at node i;

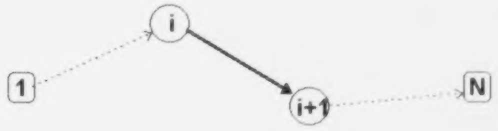


Figure 4-1 A route from origin node s to destination node g including link (i, i+1)

t = a random variable indicating the travel time on link (i, i+1). It should be noted that this variable (t) represents the link travel time under the given experiment in contrast to t_T which is a stochastic process denoting the link travel time during a whole day;

The following section will describe how the mean and variance of the route travel time can be estimated with the knowledge of the mean and variance functions of the link travel times.

4.2 THE MEAN AND VARIANCE OF ROUTE TRAVEL TIME

For each experiment described above, the total route travel time is a summation of the travel time of all the links along the route. This route travel time quantity can be obtained by calculating the arrival time at each node using a recursive formula:

$$T_{i+1} = T_i + t \tag{4-1}$$

As the estimation of the route travel times is equivalent to the estimation of the arrival time at the destination node, the arrival time has been used in the following discussion. It should be noted that estimation of the route travel time depends on how the link travel time is modeled.

In a deterministic model, the travel time on each link in the network is assumed to be an exact quantity and known a priori. In this case, the link travel time under each experiment will have the same value and thus there is only a single route travel time. In the case where the link travel time is also static, the travel time from node s to node g can be obtained by using Equation (4-1) with a constant t value for each link. It should be noted that in this case the departure time has no impact on the route travel time.

When the dynamic attribute of the link travel time is directly modeled to reflect the time-dependent attributes of the traffic network, the link travel time is a function of the time of day or the arrival time at node i (T_i). The arrival time at the destination node can be obtained by using Equation (4-1) with t replaced by a link travel time function.

However, when the link travel times are both dynamic and stochastic, and explicitly modeled by using their time dependent means and variances, the estimation of the mean and variance of the route travel time is considerably more complicated. As a simplified approach, they may be approximately estimated by assuming that the link travel times are independent between links. For comparison purposes, these models are provided as follows and are referred to in this thesis as a ***naive model:***

$$E[T_{i+1}] = E[T_i] + \mu_{(T_i)} \tag{4-2}$$

$$Var[T_{i+1}] \cong Var[T_i] + \sigma^2_{(E[T_i])} \tag{4-3}$$

The following part of this section will develop more realistic models for estimating the mean and variance of the route travel time when the link travel time is dynamic and stochastic.

4.2.1 Mean of Route Travel Time

Considering Equation (4-1), the relationship between the means of the arrival times at node i and node j is:

$$E[T_{i+1}] = E[T_i] + E[t]$$

This relationship can be further transformed as follows (Ross, 1989):

$$E[T_{i+1}] = E[T_i] + E[E[t|T_i]]$$

The recursive formula for calculating the expected travel time of the route is therefore:

$$E[T_{i+1}] = E[T_i] + E[\mu_{(Ti)}] \qquad (4\text{-}4)$$

The second term in Equation (4-4) requires the integration $\int \mu_{t(T_i)}\, f(T_i)\, dT_i$. Therefore this equation is applicable only if the PDF of the arrival time, $f(T_i)$, is available, and this means a recursive formula for estimation of the PDF of the arrival time must also be derived. This is impossible in this case because the only information available about the link travel times is their first two moments, means and variances. Furthermore, even if the PDF of the link travel time is available and the PDF of the arrival time can be derived, the integration indicated above is difficult to perform. For these reasons, an approximate recursive relationship of the arrival times is required and is presented as follows.

The main task is to identify a method to determine the second part of Equation (4-4), $E[\mu_{(T_i)}]$. Start from its definition as show in Equation (4-5):

$$E[\mu(T_i)] = \int \mu(T_i)\, f(T_i)\, dT_i \tag{4-5}$$

Assume that the mean link travel time function $\mu(T_i)$ is differentiable at point T = E[T_i], Equation (4-5) can then be expanded as a Taylor's series around this point:

$$\mu(Ti) = \mu(E[Ti]) + \mu'(E[Ti])\,(T_i - E[T_i]) + \mu''(E[T_i])\,(T_i - E[T_i])^2/2 + \ldots \tag{4-6}$$

If the series is truncated at the linear terms (or assume the second and up order derivatives are equal to zero) and then applied in Equation (4-5), the first order approximation of $E[\mu_i(T_i)]$ is obtained:

$$\begin{aligned} E[\mu(T_i)] &\cong \int\{\mu(E[T_i])+\mu'(E[T_i])\,(T_i-E[T_i])\}\, f(T_i)\, dT_i \\ &= \mu(E[T_i]) \int f(T_i)\, dT_i + 0 \\ &= \mu(E[T_i]) \end{aligned}$$

Therefore the ***first order approximation model*** of the recursive formula (4-4) is:

$$E[T_{i+1}] \cong E[T_i] + \mu(E[T_i]) \tag{4-7}$$

The first order approximation (Equation 4-7) can be successively improved by including higher order terms of the Taylor series. For example, if the second order term in Equation (4-6) is included, the second order approximation of $E[\mu_i(T_i)]$ is accordingly:

$$E[\mu(T_i)] \cong \int \{\mu(E[T_i]) + \mu'(E[T_i])(T_i - E[T_i]) + \mu''(E[Ti])(Ti - E[T_i])^2/2\}\ f(T_i)\, dT_i$$

$$= \mu(E[T_i]) \int f(T_i)\, dT_i + 0 + \mu''(E[T_i]) \int (Ti - E[T_i])^2\, f(T_i)\, dT_i /2$$

$$= \mu(E[T_i]) + \mu''(E[T_i])\, Var[T_i]/2$$

the ***second order approximation model*** of the mean arrival time can be obtained:

$$E[T_{i+1}] \cong E[T_i] + \mu(E[T_i]) + \mu''(E[T_i])\, Var[T_i]/2 \tag{4-8}$$

From above approximation models, the following observations are obtained.

REMARK 1: The first order approximation model is the same as the one from a deterministic treatment, or the naive approximation model shown in Equation (4-2). That is, the expected route travel time is found by substituting the average link travel time for the random link travel time and then calculating the route travel time. From Equation (4-8), it can also be expected that this model may be acceptable when the variance of the arrival time is very small, or the mean link travel time is approximately a linear function of time of day (i.e., $\mu''(E[T_i]) \approx 0$). This will be further examined empirically in Section 4.4.

REMARK 2: The reasonableness of the second order approximation model shown in Equation (4-8) can be illustrated using the following simple example. Consider a network with one link (i,j). Assume that the time entering the link, T_i, is random and normally distributed with a mean equal to $E[T_i]$. The link travel times are assumed to be deterministic and dynamic. There are three potential dynamic patterns: linear, convex and concave as shown in Figure 4-2. It can be found that when the link travel time is constant, the link travel time is $\mu_t(E[T_i])$ for any arrival time. Therefore, the expected link travel time will be $\mu_t(E[T_i])$, which is the same obtained from Equation (4-7) when $\mu''_t(E[T_i]) = 0$. When the link travel time is a convex function of arrival time, i.e., $\mu''_t(E[T_i]) > 0$, then the travel time under any realization of the arrival time will always be greater than $\mu_t(E[T_i])$ and therefore the expected travel time should be greater than $\mu_t(E[T_i])$. This result is compatible with the result from the Equation (4-8) because the last term in Equation (4-8), $\mu_t''(E[T_i])\ Var[T_i]/2$, is always greater than zero. A similar explanation for the case when the link travel time is concave may be demonstrated.

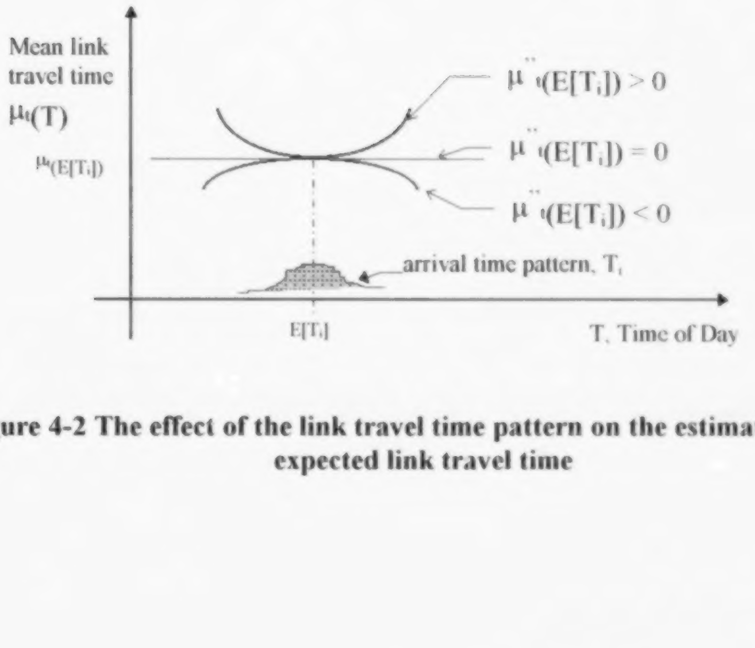

Mean link travel time
$\mu_t(T)$
$\mu_t(E[T_i])$
$\mu''_t(E[T_i]) > 0$
$\mu''_t(E[T_i]) = 0$
$\mu''_t(E[T_i]) < 0$
arrival time pattern, T_i
$E[T_i]$
T, Time of Day

Figure 4-2 The effect of the link travel time pattern on the estimation of the expected link travel time

4.2.2 Variance of Route Travel Time

The variance of the arrival time can also be derived from Equation (4-1):

$$Var[T_{i+1}] = Var[T_i] + Var[t] + 2COV(T_i , t)$$

In the above Equation, the last two parts can be estimated using following relationships (Ross, 1989):

$$\begin{aligned} Var[t] &= E[Var[t|T_i]] + Var[E[t|T_i]] \\ &= E[\sigma^2{}_{(Ti)}] + Var[\mu_{(Ti)}] \end{aligned}$$

and

$$\begin{aligned} COV(T_i , t) &= E[T_i \cdot t] - E[T_i]\, E[t] \\ &= E[E[T_i \cdot t|T_i]] - E[T_i]\, E[\mu_{(Ti)}] \\ &= E[T_i \cdot E[t|T_i]] - E[T_i]\, E[\mu_{(Ti)}] \\ &= E[T_i \cdot \mu_{(Ti)}] - E[T_i]\, E[\mu_{(Ti)}] \end{aligned}$$

Based on the above recursive formula, the arrival time variance is shown in Equation (4-9):

$$\begin{aligned} Var[T_{i+1}] = {} & Var[T_i] + E[\sigma^2{}_{(Ti)}] + Var[\mu_{(Ti)}] + \\ & 2\, E[_{Ti}\mu_{(Ti)}] - 2E[T_i]\, E[\mu_{(Ti)}] \end{aligned} \tag{4-9}$$

For the same reasons as the estimation of the mean of route travel time, the practical application of the Equation (4-9) requires an approximation method. Through a similar procedure, the first and second approximation models of the recursive Equation (4-9) can be obtained by replacing the functions $\mu(T_i)$ and $\sigma(Ti)$ with their truncated Taylor series. The ***first order approximation*** is obtained by assuming that the second and higher derivatives of $\mu(T)$ and $\sigma(T)$ are equal to zero, as shown in Equation (4-10):

$$\begin{aligned} Var[T_{i+1}] \cong \{1+ \sigma'^2(E[T_i]) + 2\mu'(E[T_i]) + \mu'^2(E[T_i])\} Var[T_i] \\ + \sigma^2(E[T_i]) \end{aligned} \tag{4-10}$$

By assuming the third and up derivatives of $\mu(T)$ and $\sigma(T)$ are equal to zero, the ***second order approximation*** of Equation (4-10) can be obtained as shown in Equation (4-11):

$$\begin{aligned} Var[Ti+1] \cong \{1+ \sigma'^2(E[T_i]) + 2\mu'(E[T_i]) + \mu'^2(E[T_i] + \\ \sigma(E[T_i])\, \sigma''(E[T_i]) - \mu''^2(E[Ti])Var[T_i]/4\} Var[T_i] \\ + \sigma^2(E[T_i]) + \mu'(E[T_i])\, \mu''(E[Ti])\, E[(T_i - E[T_i])^3] \\ + \mu''^2(E[Ti])\, E[(T_i - E[T_i])^4]/4 \end{aligned} \tag{4-11}$$

As shown in Equation (4-11), the second order approximation, although potentially leading to an improved solution, would involve identifying the third and fourth central moments of arrival time T_i. This means that in order to use this formula, a recursive formula for estimating the third and fourth order moments of the arrival times are required. This also implies that the third and fourth order moments of the link travel times need to be identified, which will bring extra complexity in the estimation procedure. On the other hand, as the arrival time at each node is a summation of many random variables (link travel times), it may be expected it would be close to a normal distribution according to the Central Limit Theorem. It is therefore recommended in this thesis that for practical purposes, the second approximation without the last two terms should be used. The new approximation formula is still called a second order approximation and is defined in Equation (4-12).

$$\begin{aligned} Var[Ti+1] \cong \{ 1+ \sigma'^2(E[T_i]) + 2\mu'(E[T_i]) + \mu'^2(E[T_i] + \sigma(E[T_i])\,\sigma''(E[T_i]) - \\ \mu''^2(E[Ti])Var[T_i]/4 \} Var[T_i] + \sigma^2(E[T_i]) \end{aligned} \tag{4-12}$$

From above approximation models, the following observations are obtained.

REMARK 3: The first order approximation model of Equation 4-10 shows that the variance of the route travel time does not depend solely on the link travel time variation, but also on the time variation pattern of the link travel times. This model can partly be verified by using a similar example as that described in Remark 2. Consider a one link situation. Assume that the time entering the link, T, may be modeled as a uniformly distributed random variable with U{a,b} (where U{a,b} represents a uniform distribution from a to b). In addition, the link travel time is assumed to be a deterministic linear function of the time of day (for example, t = k T), as shown in Figure 4-3. It is not difficult to find that the arrival time at the exit node of the link is also uniformly distributed but with different parameters, i.e., U{b+kb,a+ka}. Therefore, the variance of the arrival

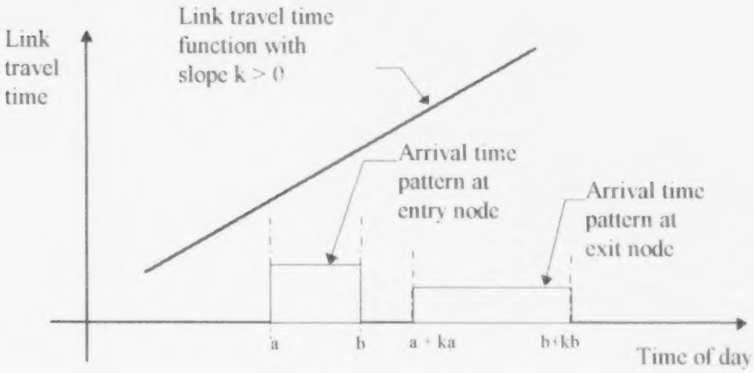

Link travel time
Link travel time function with slope k > 0
Arrival time pattern at entry node
Arrival time pattern at exit node
a
b
a + ka
b+kb
Time of day

Figure 4-3 The effect of the link travel time pattern on the estimation of the link travel time variance

time at the exiting node of the link is essentially $(1+k)^2$ times greater than the variance of the arrival time at the entering node of the link. The same conclusion can be directly obtained from Equation (4-10) as shown in Equation (4-12):

$$\mathrm{Var}[T_{i+1}] = (1+\mu'(E[T_i]))^2\ \mathrm{Var}[T_i] = (1+k)^2\ \mathrm{Var}[T_i] \tag{4-12}$$

REMARK 4: It can be anticipated that the difference between the second order approximation model (Equation 4-11) and the first order approximation model (Equation 4-10) may be trivial in applications on realistic traffic networks. The major reason for this is that the time variation of the link travel time in a traffic network is relatively small compared to the dimension of day time and hence the second order derivatives may be negligible. Section 4.4 will provide an empirical investigation of whether or not the difference is significant.

4.3 LINK TRAVEL TIME APPROXIMATION

The application of the approximation models developed above requires that the link travel time expectations associated with the random variable (both mean and variance) must be modeled as differentiable functions of time of day. Therefore, the discrete link travel time data (mean and deviation) that are typically available have to be approximated by a smooth function before they can be used. The following section discusses how the link travel time may be approximated using a differential function representing the recurring traffic congestion situation.

Under normal traffic situations, the link travel times may be assumed to be stable day by day and can therefore be statistically modeled based on historical link travel time data. This historical data may be obtained from various data sources such as road side detectors, probe vehicles, or even traffic models. Due to inherent fluctuation of traffic demand and errors in measurement related to each data source, the link travel time obtained is not a fixed value even for the same time moment of two similar days. Figure 4-4-a shows a hypothetical example of the travel time data for a link. To use these data to estimate the mean and variance of the link travel time, the time horizon is usually divided into time periods (for example, 5 min. interval) so that the number of data points for each period is high enough to provide statistically confident estimates and to minimize data management problems. As shown in Figure 4-4-b, this procedure actually models the time dependent link travel time as a step function. When the arrival time falls into a time period, the average travel time at that time period will be used. An improvement on this method is to use a pair wised linear function, as shown in Figure 4-4-c (Rilett, 1992).

In this thesis the application of the second order approximation models requires the mean and variances of the link travel time to have a second order derivative and therefore a second order polynomial is used to approximate the link travel time. If the mean travel time on link i is μ and the time entering the link is T, then the general form of the function is:

$$\mu = b_0 + b_1 T + b_2 T^2 \tag{4-13}$$

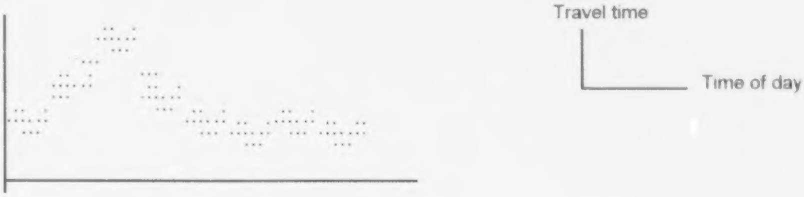
For the purpose of the route travel time estimation, the major interest of the approximation is the link travel time pattern around the mean arrival time on the link, therefore it will be neither necessary nor efficient to fit all the data with one continuous function. The approximation method developed and proposed in this thesis uses three data points. If the time in which the link is entered falls in time period k, the link travel times from time period k-1 to time period k+1 are approximated by Equation (4-13) which goes through these three points as shown in Figure 4-4-d. The parameters can be found by solving three linear Equations with three variables. If μ_k represents the mean link travel time for interval k with the middle time of the interval noted as T_k, the solution will be that of Equation (4-14):

$$\{\mathbf{b}\} = [\mathbf{T}]^{-1}\{\mu\} \qquad (4\text{-}14)$$

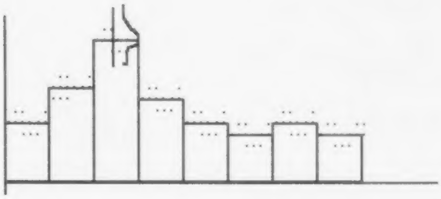
where $\{\mathbf{b}\} = \{b0, b1, b2\}'$ and $[\mathbf{T}] = \{(1, T_{k-1}, T_{k-1}^{2}), (1, T_k, T_k^{2}), (1, T_{k+1}, T_{k+1}^{2})\}'$ and $\{\mu\} = \{\mu_{i-1}, \mu_i, \mu_{i+1}\}'$

Therefore, once the arrival time is known, the link travel times (mean and variance) and their derivatives can be quickly obtained without significant extra computational burden.

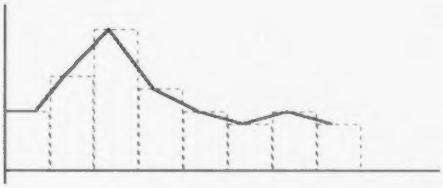
It should be noted that the approximation method proposed above is a relatively simple approach and a more comprehensive method could be used to approximate the link travel time under different situations. For example, the variance of the arrival time and the size of the link travel time interval can also be taken into account during the approximation procedure. For the situation of a smaller link travel time interval or a larger arrival time variance, more than three intervals may need to be considered in the approximation method. The underlying relation is schematically illustrated in Figure 4-5. It can be seen that a function which fits the intervals from k-2 to k+2 would be a better approximation function for the given situation. It should be noted that it won't be a problem for this thesis to incorporate this type of new approximation scheme. The only modification will be to change the function format shown in Equation (4-13) and the parameter estimation methods shown in Equation (4-14).



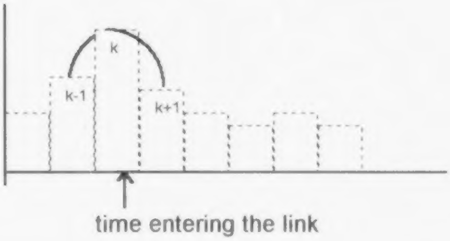
(a) Original link travel time data



(b) Link travel time approximated by step functions



(c) Link travel time approximated by composite linear functions



(d) Link travel time approximated by composite second order polynormials

Figure 4-4 Schematic illustration of link travel time approximation motheds

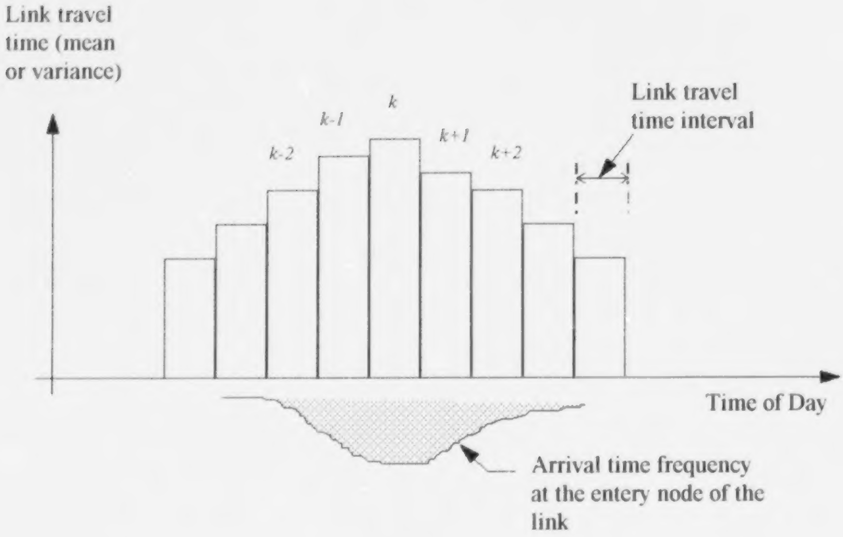

Link travel time (mean or variance)
k-2
k-1
k
k+1
k+2
Link travel time interval
Time of Day
Arrival time frequency at the entery node of the link

Figure 4-5 Illustration of the relation between the link travel time approximateing scheme with the arrival time pattern and link travel time interval

4.4 SOLUTION QUALITY OF THE APPROXIMATION MODELS: A SIMULATION STUDY

In the above section, different orders of approximation models have been derived to estimate the route travel time in a dynamic and stochastic network. Quantitatively, the second order approximation models will provide a more accurate solution than the first order models and the naive models, but they would obviously need more computational effort. However, the benefit derived from the increased accuracy will depend on the network attributes, especially the dynamic and stochastic pattern of the link travel times. This section evaluates the approximation models by comparing them with the results from a simulation model. The reason for using simulation method is twofold. One is that it is impossible, as previously discussed in Section 4.2, to obtain an explicit mathematical expression for calculating the route travel time in a dynamic and stochastic network. The other reason is that it is prohibitively expensive to carry out field experiments. Therefore, a simulation procedure is used to acquire the "real" value of the mean and variance of the route travel time. The following sections will introduce the procedure and the test network used for the simulation. The focus of the sensitivity analysis is on the relationship between the different models and various network attributes including the dynamic link travel time pattern, the link travel time variance and the link travel time distribution pattern.

4.4.1 Simulation Procedure

The purpose of the simulation procedure is to obtain the "real" route travel time PDF distribution parameters, i.e., the mean and variance of the route travel time in a dynamic and stochastic traffic network as defined in Section 4.1. The simulation procedure introduced involves repeatedly "traveling" this route with same departure time from the origin node and recording the travel times. The travel time of each trip is obtained by sequentially sampling the travel time on each link along the route. The travel time on each link has a given type of PDF with mean and variance as functions of the time entering the link. The travel time value on each of the links are sampled independently. The following section will provide further discussion on this subject. The following is the detailed procedure:

Step 1: Randomly pick i) an O-D pair in the test network, and ii) a departure time at the origin node. Select one path between the O-D and record it;

Step 2: Start from the origin node and go through all the links on the recorded path until reaching the destination node by generating the link travel time and calculating the arrival time at each node using the following formula:

$$T_{i+1} = T_i + t(T_i)$$

where T_i is the departure time at the i*th* node; $t(T_i)$ is the sampled travel time on link (i,i+1) which can be generated based on the link travel time distribution with mean $\mu(T_i)$ and variance $\sigma^2(T_i)$. The first node (or $i = 1$) is the origin node s and the departure time at this node, T_1 is given. The route travel time is the difference between the arrival time at the destination node and the origin node;

Step 3: After finishing the required iterations of step 2, the mean and variance of the route travel time for the given O-D pair and departure time are calculated by:

Calculate the mean and variance of the arrival times using (i) the naive model; (ii) the first order approximation model and (iii) the second order approximation models as presented in section 4-4.

Step 4: Goto step 1 until given number of O-D pairs are analyzed;

In this study, five hundred O-D pairs are generated and examined for each simulation combination. For each O-D pair, five thousand simulation runs are performed to obtain the "real" mean and variance of the O-D travel times. It has been theoretically estimated that this number of simulation runs will make the final estimation error smaller than 0.1 seconds for both the mean and standard deviation (Fu, 1992).

4.4.2 The Edmonton Network

A road network representing the City of Edmonton is used to test the route travel time estimation models. This network will subsequently be used in some of the following chapters. The original network data files were provided in a EMME/2 model format by the City of Edmonton. These files were then converted into the FIRST format. Figure 4-6 shows the Edmonton network from FIRST's screen output. This network, composed of 3800 links and 1400 nodes, was used for planning applications. It includes all the freeways and arterials in the Edmonton area.

4.4.3 Simulation Scenarios

For analysis purposes, the AM peak hours (6:00AM~9:00AM) were selected as the study period. Due to a lack of real time data, the dynamic and stochastic travel time patterns in the network were created based on a hypothetical change in travel time during the AM peak period. The link travel time data were then represented as a set of discrete means and standard deviations through the AM peak period. The ratios of the standard deviations to the means, i.e., the coefficient of variation (COV), is assumed to be constant.

In order to analyze the sensitivity of the solution quality of the approximation models to the network attributes, three types of network attributes were investigated:

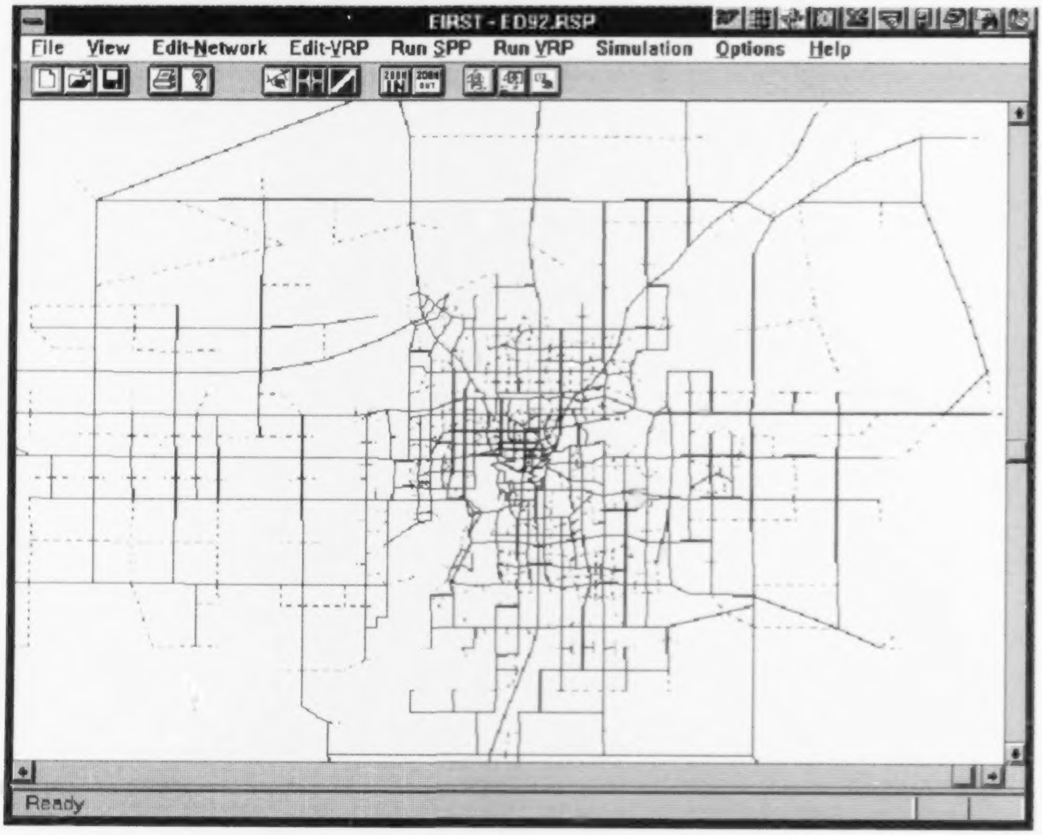

FIRST - ED92.RSP
File
View
Edit-Network
Edit-VRP
Run SPP
Run VRP
Simulation
Options
Help
ZOOM IN
ZOOM OUT
Ready

Figure 4-6 The Edmonton Network

1. Dynamic pattern of the link travel time during the AM peak period: Three types of dynamic link travel time patterns were created and are respectively labeled as Peak Pattern A, Peak Pattern B and Peak Pattern C as shown in Figure 4-7. Peak Pattern A represents a three hour peak time period (from 6AM to 9AM) while Peak Pattern B represent a two hour peak time period (from 6AM to 8AM). Peak Pattern C has a peak time period from 6AM to 7AM. It should be noted that these three peak patterns are not necessary real representations of the traffic patterns, however they can be interpreted as representations of the time variation of traffic pattern from a relatively smooth situation (Pattern A) to a relatively peaked situation (Pattern C);

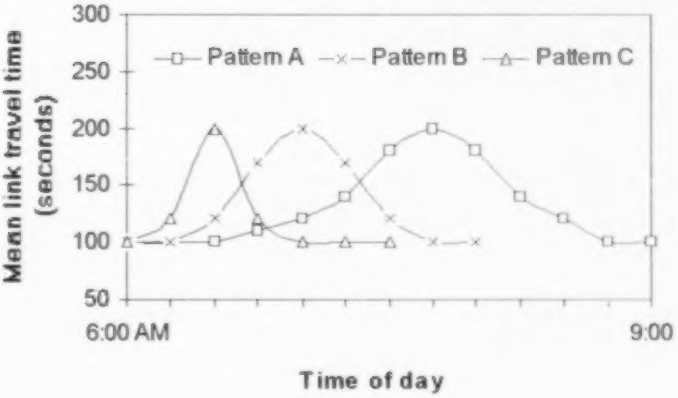


Figure 4-7 Time variation patterns of the link travel time

2. Travel time coefficient of variation (COV): The COV of the travel times on each link is assumed to range from 0.0 to 1.0. Three discrete values, 0.1, 0.5 and 0.9 are used in the simulation study;

3. Travel time PDF: As discussed in Chapter 3, the link travel time distribution could be very complicated and is impossible to be modeled as a single distribution model such as a normal distribution or a log normal distribution. This simulation examined three scenarios. As shown in Figure 4-8, the first scenarios assumes that

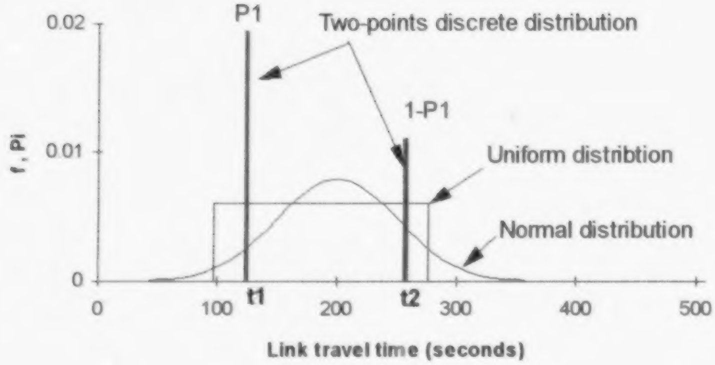


Figure 4-8 Probability distribution patterns of the link travel time

all links have normally distributed link travel times. The next scenario assumes that each link has a uniformly distributed travel time. The last scenario assumes that each link has only two possible values, t1 and t2 with probability P1 and 1-P1 respectively as shown in Figure 4-8. This last scenario can be viewed as an extreme example of bimodal distribution. For the comparison purpose, it was assumed that for each link the travel time mean and variance under three scenarios are same. It should be noted that for both normal distribution and uniform distribution their distributions can be uniquely decided by mean and variance. For the two-value discrete distribution, it was further assumed that the probability P1 and thus 1-P1 is known, and consequently the t1 and t2 value can be calculated based on the mean and variance.

4.4.4 Approximation Quality and Sensitivity Analysis

4.4.4.1 General performance

This section demonstrates the general performance of the approximation models under a given link travel time pattern (Pattern B) and a given link travel time COV (COV=0.5). The link travel times are assumed to have normal distributions. Figure 4-9 shows the relationship between the approximation error of the first order model and the second order model for estimating the mean of the route travel times. The approximation error is defined as the absolute difference between the approximated result and simulated result and the following sections will use the same definition without further notation. It can be seen that the estimation error for the second order model is primarily less than 15 seconds while it is primarily less than 25 seconds for the first order model. However, these average estimation errors are not so significant if compared to the average trip time of 1840 seconds.

Figure 4-10 and Figure 4-11 show the scatter graph of the route travel time standard deviations calculated by the approximation models (the naive model and the first order approximation model) and by the simulation method. It can be seen that the improvement in using the first order model is significant compared to the naive model. The average estimation error is 3 seconds for the first order model as compared to 28 seconds for the naïve model.

The simulation results (Figure 4-12) also show that the difference between the first order model and second order model for estimating the route travel time variance is insignificant. This results confirm Remark 4 in Section 4.2

4.4.4.2 Sensitivity to the link travel time COV

The purpose of this section is to examine how the variations of the link travel times in the network influence the solution quality of the approximation models. The simulation uses a single link travel time pattern (Pattern B) and a single probability distribution (normal distribution) for all links in the network. Figure 4-13 shows the relationship between the approximation error of the first order model and the second order model for estimating the mean of the route travel times as a function of link travel time COV. As would be expected, the improvement of the second order model over the first order model is related to the magnitude of the link travel time variance. For example, when the COV is doubled the second model has approximately one half the error as the first order model.

Figure 4-14 shows the graph of the estimation error of the route travel time standard deviations calculated by the approximation models (the naive model and the first order approximation model) as a function of the link travel time COV. It can be seen that the improvement of using the first order model or second order model over the naive model is more significant than that found from the mean route travel time estimation sensitivity analysis. This would be expected because the rate of change of the average link travel time influences the route travel time variance calculation as shown in Remark 3 of Section 4.2.

4.4.4.3 Sensitivity to the dynamic pattern of the link travel time

This section shows how the time variation pattern of the link travel time in the network influences the solution quality of the approximation models. The link travel times are normally distributed with the COV equal to 0.5. Figure 4-15 and Figure 4-16 show the relationship between the estimation errors of approximation models and the time variation pattern of the link travel times. Similar to the finding in the previous sections, the improvement of using the second order model over the first order model for estimation of mean route travel time (Figure 4-15) and the first order model over the naïve model for estimation of route travel time variance (Figure 4-16) is significant.

As would be expected, the route travel time estimation error is highly correlated to the dynamic pattern of the link travel times. The more peaky the link travel time pattern, the higher the estimation error. It can also be found that the estimation of the mean route travel time is much more sensitive to the estimation of the route travel time variance. This is because the accuracy of the mean route travel time model mainly depends on the second order derivative of the link travel time which is significantly influenced by the change rate of the link travel time pattern. Similar to the results in the previous sections, the improvement of using the second order model over the first order model for estimation of mean route travel time is apparent. The relative improvement of the second order model over the naïve model for estimation of route travel time variance is negligible. However, both these models are significantly more accurate than the naïve model.

It can also be seen that the improvement of the higher order models over the lower order models is insignificant as the link travel times in the network changes more sharply (from pattern C to pattern A). This result is expected for the reasons explained in Remark 4 previously discussed in Section 4.2.2.

4.4.4.4 Sensitivity to the PDF of the link travel time

This section shows how the link travel time PDF in the network influences the solution quality of the approximation models. The COV of the link travel times are set to 0.5 and Pattern B is used to represent the dynamic pattern of the link travel times. Figure 4-17 and Figure 4-18 show the relationship between the estimation errors of the approximation models and the link travel time distribution.

It can be found that the link travel time distribution pattern has no significant effect on the estimation of both the mean and variance of the route travel time. Based on this empirical result, it may be concluded that the simple normal distribution could be used for route travel time estimation purposes. This also implies that the relevant information on the link travel time are its mean and variance as a function of the time of day.

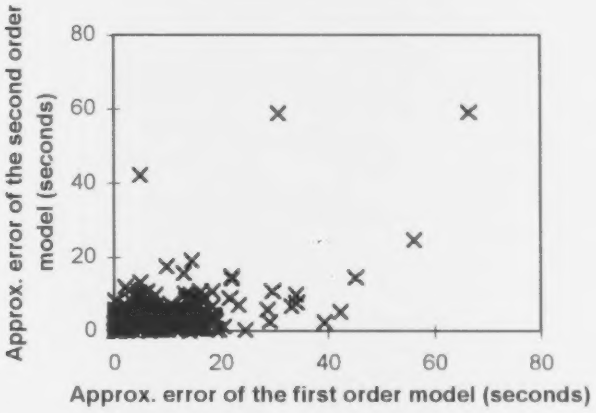


Figure 4-9 A Comparison of the first order model with the second order model

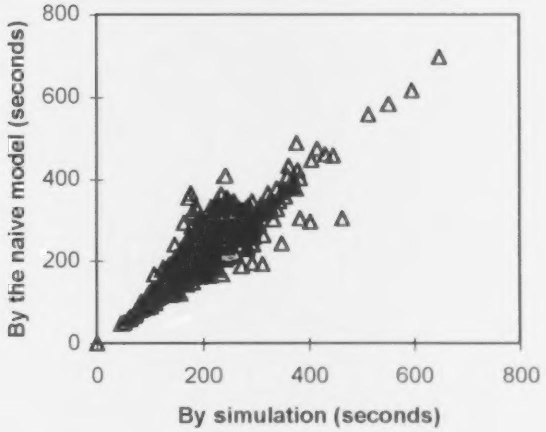


Figure 4-10 Estimation of route travel time standard deviation: naive approximation model vs. simulation results

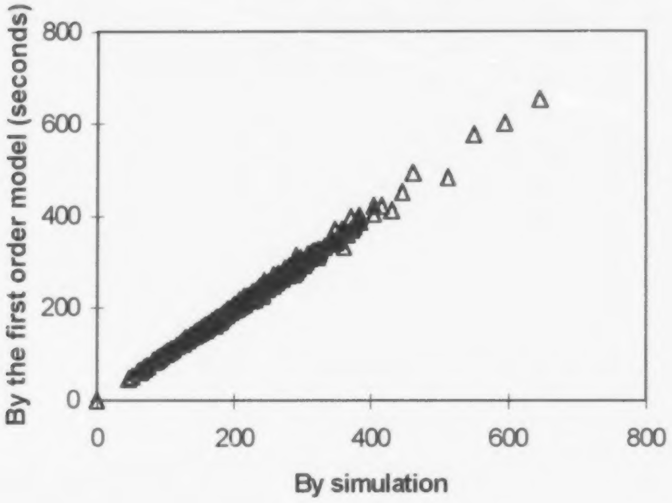


Figure 4-11 Estimation of route travel time standard deviation: the second order approximation model vs. the simulation results

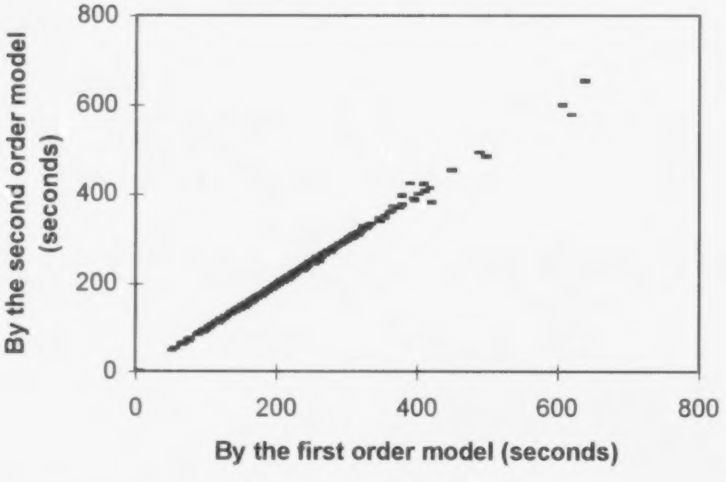


Figure 4-12 Estimation of route travel time standard deviation: the first order model vs. the second order model

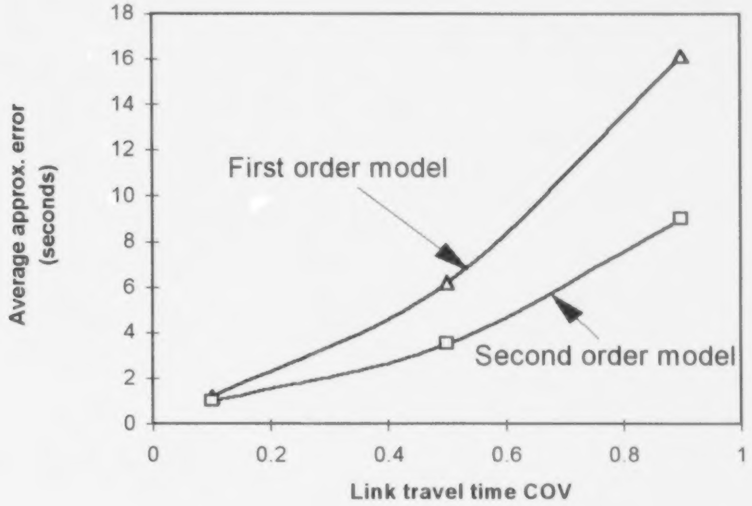


Figure 4-13 Relationship between the route travel time mean estimation quality with link travel time covariance

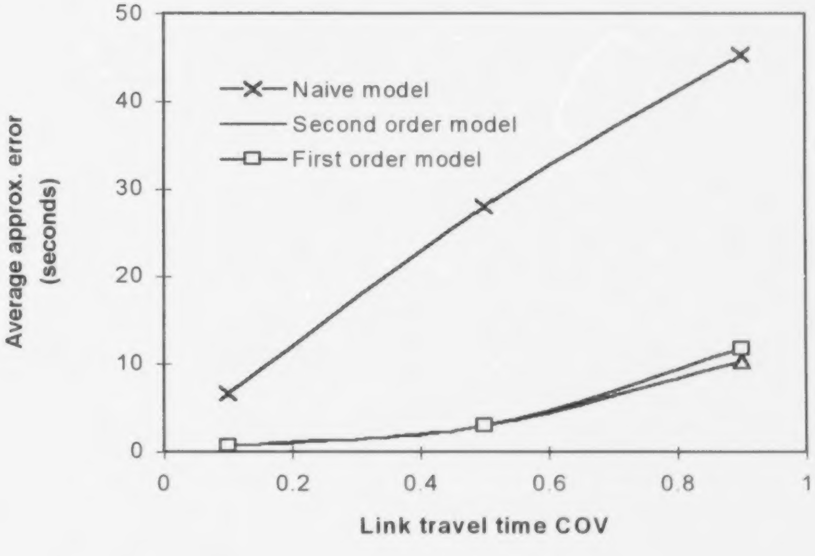


Figure 4-14 Relationship between the route travel time standard deviation estimation error with link travel time covariance

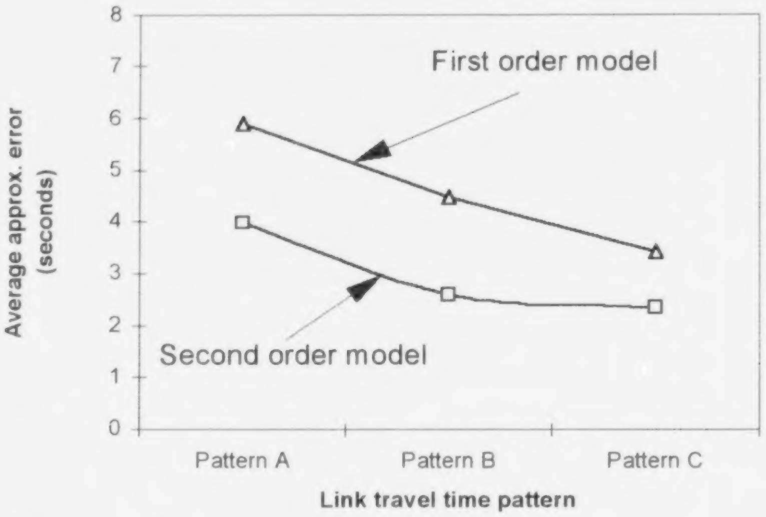


Figure 4-15 Relationship between the route travel time mean estimation error with link travel time pattern

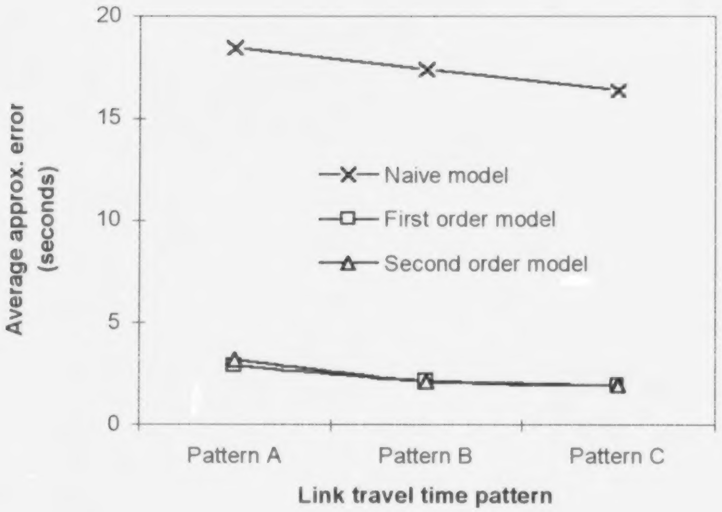


Figure 4-16 Relationship between the estimation error of the route travel time standard deviation with link travel time pattern

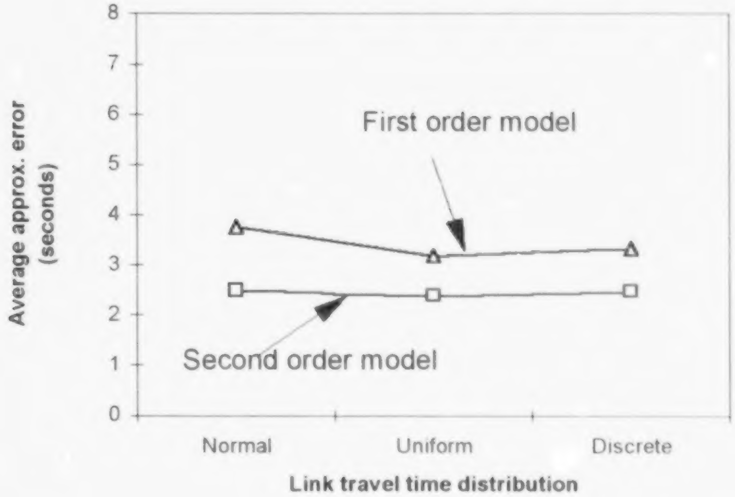


Figure 4-17 Relationship between the route travel time mean estimation quality with link travel time distribution

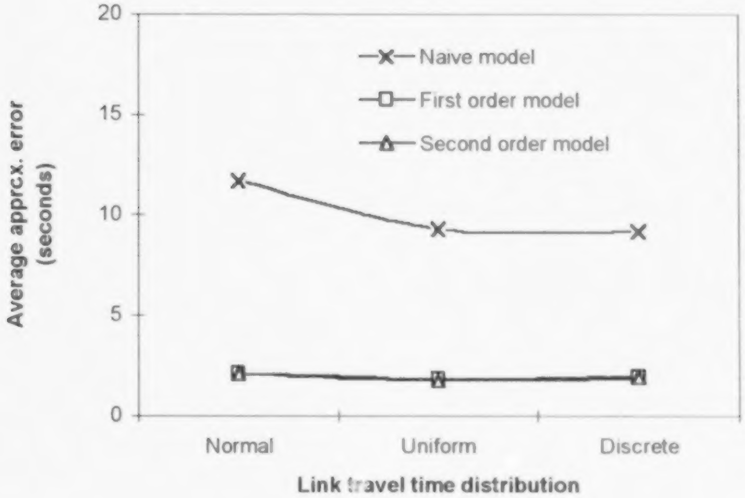


Figure 4-18 Relationship between the route travel time variance estimation quality with link travel time distribution

4.5 CONCLUSIONS

This chapter developed several approximation models to estimate the mean and variance of the route travel time in traffic networks where the link travel times are dynamic and stochastic. The dynamic and stochastic attributes of the link travel times are modeled by the mean and variance of the link travel time as a function of time day or arrival time at the link. The approximation models are derived and examined theoretically and empirically. The major conclusions are summarized as follows:

(1). On the estimation of the expected route travel time:

- In a dynamic and stochastic network, the expected route travel time is not a summation of the expected link travel times. It at least also depends on the average link travel time pattern during the time of day (for example, the second order derivative of the expected link travel time) and the link travel time variations. The traditional method (the naive model) may overestimate or underestimate the expected route travel time.
- The expected route travel times can be more accurately estimated using the second order approximation model. Based on the simulation study, the relative improvement of the second order approximation models over the first order model (or naive model) is quite small (less than 0.2%). However, this small percentage of difference could be important for route selection in a traffic network.
- The relative improvement of the second order model compared to the first order model or naive model depends on the link travel time variance and the severity of

the time variation of the link travel time. The relative improvement becomes greater as the link travel time variances and the link travel time changing rate increase.

- The link travel time distribution pattern has no significant effect on the mean route travel time estimation. Therefore, the normal distribution may be used to represent the link travel time distribution.

(2). On the estimation of route travel time variance:

- In a dynamic and stochastic network, the route travel time variance does not eqaul to the summation of the link travel time variances. It also depends on the link travel time changing pattern during the time of day, for example, the first and second order derivatives of the expected link travel time and link travel time variance. The traditional method (naive models) may overestimate or underestimate the route travel time variance.

- The variance of the route travel times can be more accurately estimated using the first or second order approximation models. Based on the simulation study, the first order model and the second order model are quite close in terms of solution quality, however, they provide significantly better solutions than the naive model.

- The relative improvement of the first or second order models compared to the naive model depends on the link travel time variance and the severity of the time variation of the link travel time. The relative improvement becomes greater as the link travel time variances and the link travel time changing rate increase.

REFERENCES:

Dreyfus, E. S. (1969),"An Appraisal of Some Shortest Path Algorithms," *Operations Research.* **17**, 395~412.

Fu, L. (1992), Determination of Sample Size For Transportation Planning, Working paper for Course CIVE 613, University of Alberta, Edmonton.

Kaufman E. D. , J. Lee, R. L. Smith, (1990), "Anticipatory Traffic Modeling and Route Guidance in Intelligent Vehicle-Highway Systems," *IVHS Technical Report-90-2*, University of Michigan.

Rilett, L. R. (1992), *Modeling of TravTek's Dynamic Route Guidance Logic Using the Integration Model,* Ph.D. Dissertation, Queen's University, Kinston, Ontario.

Ross M. S. (1989), *Introduction to Probability Models, Third Edition,* Academic Press Inc. San Diego, CA.

CHAPTER 5

ESTIMATION OF EXPECTED MINIMUM PATHS IN DYNAMIC AND STOCHASTIC TRAFFIC NETWORKS[*]

5.0 INTRODUCTION

For most RGS currently under development the optimal route between an origin and destination is defined as the one with a minimum expected travel time. This optimal route is commonly calculated by applying a Dijsktra type shortest path algorithm where the link travel times are treated deterministically rather than stochastically. Typically the random link travel times are replaced by average link travel time values for a set of discrete time intervals throughout the day. The drawback to this type of deterministic treatment is that while it makes the problem computationally tractable it may in fact generate sub-optimal solutions of the problem as shown in this chapter.

Conversely, when both the dynamic and stochastic nature of link travel times are explicitly modeled, the optimal algorithms can become computationally inefficient and/or impractical for use within an actual application. The objective of this chapter is to develop a new shortest path algorithm which can provide improved solutions without significantly adding to the overall computation time. Essentially, this chapter attempts to answer the question: how can the uncertainty associated with link travel times be incorporated within the calculation of the expected optimal routes?

---

[*] A modified version of this chapter has been published in "Proceedings of Vehicle Navigation and Information System (VNIS)", 1995 Annual Meeting of VINS, Seattle, Washington.

The shortest path problem has been studied extensively in the fields of computer science, operations research and transportation engineering. Most of the literature has focused on the problem in which the link travel cost (or weight) is assumed to be static and deterministic. Many efficient algorithms have been developed (Bellman, 1958; Dijsktra, 1959; Dreyfus, 1969) and in this thesis, these algorithms are referred to as the standard shortest path algorithms. It should be noted that the standard shortest path algorithms also have been found to be applicable under certain conditions in dynamic networks where the deterministic link travel time is a function of the time of day (Cooke and Halsey, 1966; Kauman and Smith, 1990).

Frank (1969) and Mirchandani (1976) first studied the problem of determining the probability distribution of the shortest path length in a stochastic network where link travel times are random variables. Loui (1983), Mirchandani and Soroush (1986) studied the shortest path problem with different types of utility functions. It was found that if the objective is to identify the expected shortest path, then the problem simply reduces to a deterministic shortest path problem in a network where the random link travel times are replaced by their expected values. Therefore, the efficient standard shortest path algorithms can still be used to find the expected shortest paths in a static and stochastic network.

In a dynamic and stochastic network, the link travel costs in general manifest a time-dependent pattern (dynamic), but are not deterministic at any point in time or time interval. This situation is a more realistic representation of a traffic network because the travel times on traffic networks generally change with time of day in some general pattern (for example, peak and off-peak periods) which has a certain amount of variation associated with it. However, the standard shortest path algorithm may fail to find the expected shortest path in dynamic and stochastic networks as demonstrated by Hall (1986). An algorithm was proposed to find the optimal route and this algorithm was demonstrated on a small transit network example. The algorithm however can only be practically applied in solving problems on small networks because of computational constraints.

This chapter is organized as follows. The dynamic and stochastic shortest path problem is first defined and the properties associated with this problem are discussed. A heuristic algorithm based on the k-shortest path algorithm is subsequently proposed. Finally, the tradeoffs between solution quality and computational efficiency of the proposed algorithm will be demonstrated on a realistic network from Edmonton, Alberta.

5.1 DYNAMIC AND STOCHATIC SHORTEST PATH PROBLEM: DEFINITION AND PROPERTIES

As discussed previously in Section 4.1, a dynamic and stochastic network is defined by assuming that the link travel times on some or all of the links in the network are random variables and their probability distributions are dependent on the time of day. Furthermore, it is assumed that the link travel times are continuous random variables and the only available information about their distribution is their respective means and variances. The problem is to find the optimal path, that is, the path with the minimum expected travel time from an origin to a destination with a given departure time in the network. This problem is referred to as the dynamic and stochastic shortest path problem (DSSPP).

In the previous chapter, some approximation models were developed for estimating the mean and variance of the route travel time in a dynamic and stochastic network. These approximation models (the second order approximation model for the mean and the first order approximation model for the variance) will be further used to analyze the basic attributes or properties of the DSSPP in the following discussion. To facilitate the discussion, these two equations are restated below (Figure 5-1):

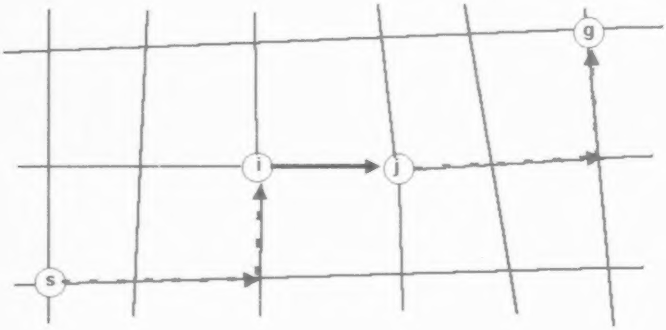


Figure 5-1 A path from origin node s to destination node g including link (i, j) in a traffic network

$$E[T_j] \cong E[T_i] + \mu(E[T_i]) + \mu''(E[T_i])\, Var[T_i]/2 \qquad (5\text{-}1)$$

$$Var[T_j] \cong \{1 + \sigma'^2(E[T_i]) + 2\mu'(E[T_i]) + \mu'^2(E[T_i])\} Var[T_i] + \sigma^2(E[T_i]) \qquad (5\text{-}2)$$

where,

T_i = a random variable indicating the arrival time or departure time at node I,

$E[T_i]$ = the expected arrival time at node i;

$Var[T_i]$= the variance of the arrival time at node i;

$\mu(T)$ = the mean travel time on link (i,j) as a function of time of day, T, and μ' and μ'' represent respectively the first and second order derivatives of μ;

$\sigma(T)$ = the standard deviation of the travel time on link (i,j) as a function of time of day, T, and σ' and σ'' represent the first and second order derivatives of σ;

From Equation (5-1) and (5-2), the following properties of the DSSPP may be observed:

PROPERTY 1: If the mean link travel time as a function of time ($\mu(T)$) of at least one link in a network is non-linear, the standard shortest path algorithms may fail to find the optimal path between two nodes in the network.

This observation may be illustrated by the use of the example network shown in Figure 5-2. The network is composed of two sub paths (p1 and p2) from the origin node s to an intermediate node i, and one link (i, j) from node i to the destination node j.

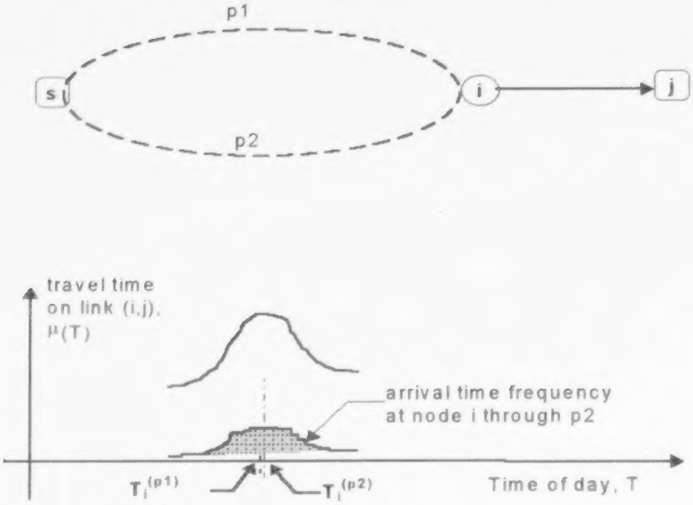

p1
s
i
j
p2
travel time on link (i,j), μ(T)
arrival time frequency at node i through p2
$T_i^{(p1)}$
$T_i^{(p2)}$
Time of day, T

Figure 5-2 A simple dynamic and stochastic network

Assume that the travel time on p1 is deterministic and that the travel time on p2 is stochastic. The travel time on link (i,j), $\mu_{(T)}$, is deterministic but changes with time in a non-linear fashion as shown in Figure 5-2. If the expected arrival time at node i through p1, $T_i^{(p1)}$, is marginally less than through p2 ($T_i^{(p2)}$) then the subpath p1 is the minimum expected route from node s to node i. On other hand, it can be seen in Equation (5-1) that the expected minimum arrival time at node j not only depends on the expected arrival time at node i, but also on the variance of the arrival time at node i and the second derivative of the mean travel time on link (i, j). Given that the travel time on link (i, j) is concave and hence its second derivative is negative, it is possible that the subpath p2 is on the expected minimum path from s to j. In short, Bellman's "principle of optimality" which states that any subpath of a shortest path must be a shortest path (Denardo,1982), does not hold in a DSSPP.

The above property also implies that the standard shortest path algorithms could be applicable if the link travel time function $\mu(T)$ is linear or in a more relaxed sense, close to linear within the local range.

PROPERTY 2: The DSSPP is intractable.

The intractablity of the DSSPP can be shown using a special acyclic type of network (Garey and Johson, 1979). As shown in Figure 5-3, the network has N nodes sequentially labeled from 1 to N, with 2 links between each successive pair of nodes. The problem is to find the expected shortest path from node 1 to node N. If the network is deterministic, the problem can be solved by finding the shortest path to node 2 first, then node 3, and onward until node N. The computation time is O(n). This type of procedure will not work in a dynamic and stochastic network because the optimal path to node i does not have to be part of the optimal path to node i+1. This means that all the 2^{N-1} paths from 1 to N must be examined before the optimal path can be definitely identified. The computation time therefore grows exponentially with the number of nodes, N.

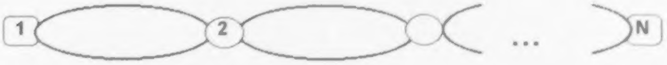


Figure 5-3 An acyclic network

5.2 A HEURISTIC ALGORITHM TO CALCULATE THE EXPECTED SHORTEST PATH

In the previous section it was shown that standard shortest path algorithm may not identify the expected shortest path on a dynamic and stochastic network and the DSSPP is intractable in the sense that there is no polynomial time algorithm, like the standard shortest path algorithm, to solve this problem. Therefore, a heuristic algorithm was developed to identify the "optimal" routes.

As shown in Chapter 4, it would be expected that under non-incident conditions the rate of change of the mean link travel time would be relatively small. Therefore, the path identified by the standard shortest path algorithm would typically be close to the expected shortest path. The heuristic algorithm proposed in this thesis uses this fact to identify the best route without significant additional computation efforts. The algorithm is based on the k-shortest path algorithm and has a parameter K indicating that K shortest paths will be examined. The algorithm proceeds as follows:

***Step 1*:** Find the shortest, the second shortest and up to the K*th* shortest paths from the origin node to the destination node, based on the mean travel times over links in the network. These are stored in ascending order in list A.

***Step 2*:** Set k = 1; take the k*th* shortest path from A and call it P. Calculate the expected travel time over P by using Equations (5-1) and (5-2), denoted it L_{opt}.

***Step 3*:** If k > K: P is the "optimal" path, L_{opt} is the minimum expected travel time, Stop.

Otherwise, go to step 4;

***Step 4*:** Set k = k+1, take the k*th* shortest path from A and call it P_k. Calculate the expected travel time over P_k by using Equations (5-1) and (5-2) and denoted it L_k

If $L_k < L_{opt}$: $P = P_k$ and $L_{opt} = L_k$ and Goto step 3.

There were three issues that needed to be addressed before this algorithm could be implemented. The first issue was to identify the technique for finding the K shortest paths. This thesis uses Shir's k-shortest path label setting algorithm due to its close relation with the shortest path label setting algorithms (Shir,1979).

The second issue is to identify the value of K. From a practical point of view the appropriate K value can be based on an empirical sensitivity study. The use of larger values for K will increase the chances of finding the optimum expected shortest path, but at same time will require a greater computational effort. The balance of solution quality and computation cost in determining the K value will be further discussed in Section 5-3.

Finally, the proposed heuristic algorithm requires applying the approximation formulae in Equations (5-1) and (5-2) which are derived based on the assumptions that the mean and standard deviation of the link travel time are continuous functions of time of day and have at least second order derivatives. This can be achieved using the link travel time smoothing method as discussed in the section 4.4 of Chapter 4.

5.3 COMPUTATIONAL ANALYSIS

The objective of this section is to demonstrate the solution quality and computational efficiency of the proposed algorithm with respect to the value of the parameter K used in the algorithm. The heuristic algorithm developed in this thesis was coded in C++ and executed under the Microsoft Windows operating environment on a 486 compatible with 50 MHz speed and 8 MB RAM.

The experiment was performed on a network from the City of Edmonton. This network, composed of 3800 links and 1400 nodes, is primarily used for planning applications. The AM peak (6:00AM~9:00AM) was selected as the study period. Due to a lack of real-time data, the dynamic and stochastic travel time patterns in the network were created based on the free flow speed on each link and a theoretical change in travel time during the AM peak period. The link travel time data were then represented as a set of discrete means and standard deviations through the AM peak period. The standard deviation range is from 10 to 20 percentage of the mean travel time. Figure 5-4 shows an example of the means and standard deviations of the travel times on an example link.

To demonstrate the solution quality of the heuristic algorithm a comparison is usually made with the optimal solution. Because it is too computationally intensive to identify the expected optimal path in a large network, a reference path is used for comparison purposes. In this thesis the reference path is set as one found using the proposed algorithm with a pre-specified value of K. A K value of 10 was used indicating that the best path within these 10 paths is the "optimal" path. The relationship between the solution and the K value will be discussed in the following paragraphs

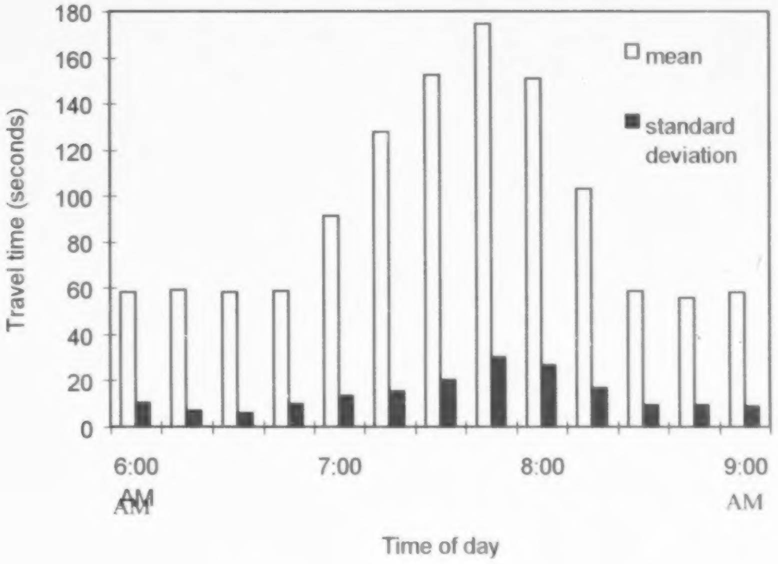


Figure 5-4 Link travel time pattern

Three hundred O-D pairs were randomly generated and their respective expected minimum paths were calculated using the proposed algorithm. Figure 5-5 shows the relationship between the K value and the percentage of time the optimal path was found. For example, when the K value is equal to one, there is a 30% chance that the minimum path route would not be identified whereas if K is increased to 5 this percentage decreases to 5%. It should be note that a K value of 1 corresponds to the case of simply using a standard minimum path algorithm.

Figure 5-6 shows the relative error of the solution as a function of the K value. It can be seen that the relative error is very small. For example, when K is equal to 1, the relative error is very small (less than 0.3%). The average absolute error for a K value of 1 is approximately 5 seconds with a maximum error of 120 seconds with respect to the average travel time of 1788 seconds. It may be seen in Figures 5-5 and 5-6 that the greatest jump in accuracy occurs at the lower K values (i.e. more improvement from K=1 to 2 than from K = 9 to 10).

The computation time of the proposed algorithm with respect to the K value is shown in Figure 5-7. It can be seen that the CPU time increase is fairly significant. For example, when K is equal to 2 the increase in CPU time is approximately 90%. However, it should be kept in mind that this algorithm is considerably faster than a complete enumeration.

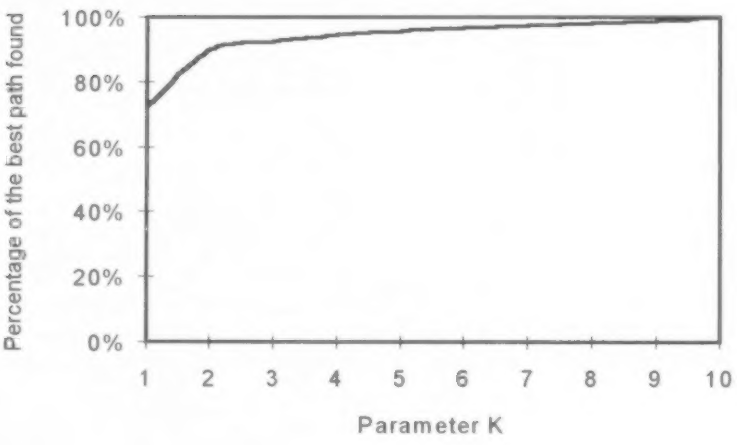


Figure 5-5 Solution quality vs. K value

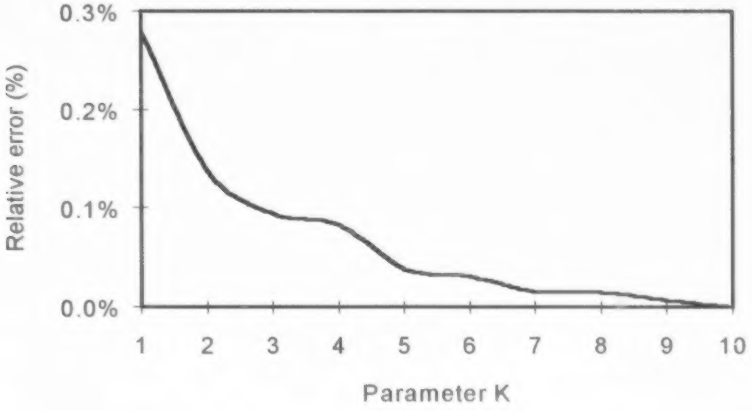


Figure 5-6 Solution quality vs. K value

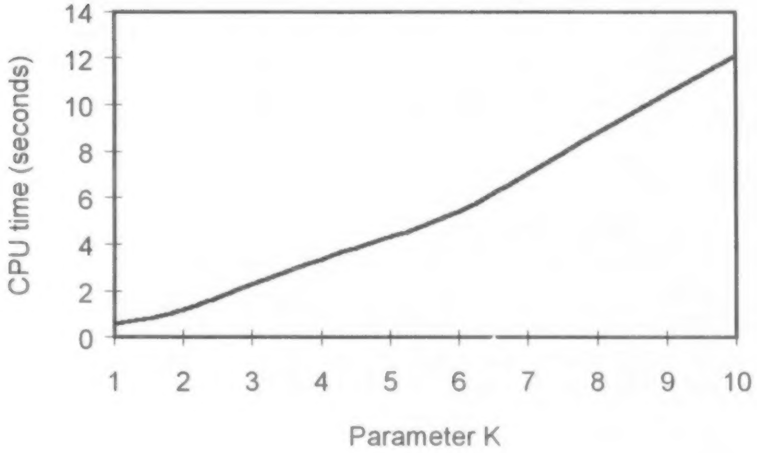


Figure 5-7 CPU time vs. K value

5.4 CONCLUSIONS

It was shown that the DSSPP is computationally intractable and that it cannot be solved exactly using standard shortest path algorithms. This thesis proposed a heuristic algorithm for solving the DSSPP where the dynamic and stochastic attributes of the link travel times are modeled by the mean and variance of the link travel time as a function of time of day. The algorithm is based on k-shortest path algorithm and its performance was tested in a real size network. The following points were illustrated in this study:

- The standard shortest path algorithms may fail to find the minimum expected paths in a dynamic and stochastic network. The solution error by the standard shortest path algorithm was shown to be relatively small (5 seconds on average) primarily because of the simplicity of the network and , more importantly, because the dynamic travel times changed relatively slowly with time. It is anticipated that a greater impact would be found during incident conditions.
- The proposed heuristic algorithm provided improved solutions with only a moderate increase in overall computation time. The solution improvement was found to be significant with the K value increasing from 1 to 2 (more than 18% in terms of the percentage of finding the best solution although the increase in computation time was on the order of 90%);
- While theoretically incorrect, the use of standard shortest path algorithms in dynamic and stochastic traffic networks may be applicable from a practical perspective. This will be especially true if the change of travel time in the network is moderate.

- Finally, it should be noted that the above conclusions are based on hypothesized link travel time data. It would therefore be necessary to conduct further studies based on real travel time data before any general conclusions are made. It would also be beneficial to conduct the experiments during incident conditions to identify whether this technique has any potential benefits in these situations.

REFERENCES:

Bellman, R. E. (1958), "On a Routing Problem," *Quarterly of Applied Mathematics.* **16**, 87~90.

Denardo, E., *Dynamic Programming: Models and Applications*, Prentice-Hall,Inc, N. J., 1982.

Denardo, V. E. (1982), Dynamic Programming: Models and Applications, Prentice-Hall,Inc, N. J.

Dreyfus, E. S. (1969), "An Appraisal of Some Shortest Path Algorithms," *Operations Research.* **17**, 395~412.

Gary, R. M. and D. S. Johnson*: Computers and Intractability: a Guide to the Theory of NP-Completeness*, Freeman, S Francisco 1979.

Glove, F., D. Klinkgman and N. Philips, (1985), "A new Polynomially bounded Shortest Path Algorithm," *Operations Research*, Vol. **33**, 65~73.

Hall,R. (1990), "The Fastest Path Through a Network with Random Time-Dependent Travel Time," *Transportation Science.* Vol. **20**, No. 3, pp. 182~188.

Kaufman, E. D., J. Lee and R. L. Smith, (1990), "Anticipatory Traffic Modeling and Route Guidance in Intelligent Vehicle-Highway Systems," *IVHS Technical Report-90-2*, University of Michigan.

Loui, R. P. (1983), "Optimal Paths in Graphs with Stochastic or Multidimensional Weights" Comm. ACM, 26, 1983, 670~676.

Mirchandani, P. and Soroush, H. (1986), "Routes and Flows in Stochastic Networks" in Advanced Schools on Stochastic in Combinatorial Optimization, edited by Angrealtta, G., Mason, F. and Serafini, P.

Pape, U. , (1974), "Implementation and Efficiency of Moore Algorithms for The shortest Route Problem," *Mathematical Programming* 7, 212~222.

Shier, R. D., (1979), "On Algorithms for Finding the k Shortest Paths in a Network," *Networks*, Vol. **9**, 195~214.

Tarjan, E. R., *Data Structure and Network Algorithms*. SIAM, Philadelphia, Pa, 1983.

CHAPTER 6

HEURISTIC SHORTEST PATH ALGORITHMS

6.1 INTRODUCTION

One of the major requirements for in-vehicle Route Guidance System (RGS) and real-time Automated Vehicle Dispatching System (AVDS) is the ability to calculate the shortest path from an origin to a destination in a quick and accurate manner. In a distributed RGS, an in-vehicle computer is commonly used to calculate the optimal route in a large traffic network. Typically the recommended routes must be found within a very short time period (e.g., a few seconds). In an AVDS, due to real-time operational requirements, new routes and schedules must be determined within a reasonable time after a customer requests a service. Because the O-D travel times, which are the basic input to the routing and scheduling procedure, are dynamic in an urban traffic environment, a shortest path algorithm has to be repeatedly used to calculate the O-D travel times during the optimization procedure. More detailed discussion will be provided in Chapter 8 as for how the O-D travel time estimation method influences the efficiency of the dial-a-ride vehicle routing and scheduling process.

In the above applications, optimal shortest path algorithms typically cannot be directly used because they are too computationally intensive to be feasible for real-time operations (Kuznetsov,1993). The consideration of the dynamic and stochastic pattern of traffic networks imposes extra computational burden to solve the shortest path problem, as previously discussed in Chapter 5. The objective of this paper is to develop heuristic shortest path algorithms using various heuristic search strategies from the field of Artificial Intelligence (AI) and to demonstrate their solution quality and computation efficiency when these algorithms are implemented in traffic networks. For this purpose, the traffic network is assumed to be static and deterministic in our discussion. However, it should be noted that most proposed heuristic algorithms work equally efficient when they are applied in dynamic and stochastic traffic networks (when the traditional shortest path algorithm is applicable for the reason presented in Chapter 5).

Heuristic search strategies have traditionally been investigated by researchers in the AI field (Hart *et al.* 1968; Nilsson 1971; Newell *et al.* 1972; Pearl, 1984). The shortest path problem is typically used as a testing mechanism to demonstrate the effectiveness of these heuristics. Because the performance of a given heuristic algorithm is a function of the particular application or network, the conclusions from one application usually cannot be generalized to others.

The current RGS field tests in North America, Europe and Japan have generated renewed interest in using heuristic algorithms to find shortest paths in a traffic network for real-time vehicle routing operations. Guzolek and Koch (1989) discussed how some heuristic search methods can be used in a vehicle navigation system, but there has not been an comprehensive study examining the implementation and performance of the algorithms. Kuznetsov (1992) discussed the application of the algorithm A* (called the force driven method in his paper), bi-directional search methodology and hierarchical search methodology used for pathfinding in the TravTek project. Although some empirical results have been presented, no exact information has been provided on the algorithms themselves.

This chapter is organized as follows. A brief overview on the optimal shortest path algorithms and their computational performance are first provided because they are the benchmark that the proposed heuristic algorithms will be compared to. Next, this chapter specifically examines three heuristic search strategies that may be classified as i) Limiting the search area; ii) decomposing the search problem, and iii) Limiting the search links. In addition, new heuristic shortest path algorithms are developed by combining the optimal shortest path algorithms with the heuristic search strategies. The algorithmic implementation of the proposed heuristic algorithms are detailed. Finally, the computational efficiency and solution quality of the new heuristic algorithms are demonstrated on a network from Edmonton, Alberta.

6.2 THE SHORTEST PATH PROBLEM AND OPTIMAL ALGORITHMS

A road traffic network is represented by a digraph G(N,A) that consists of a set of nodes N and a set of arcs A (or links used in this thesis). Denote the number of nodes $|N|=n$ and the number of links $|A|=m$. A link $a=(i,j)\in A$ is directed from node i to node j and has an associated general cost c_{ij}. The general cost represents the impedance of an individual vehicle going through that link and is usually described by link travel time, link length and toll fee, or some combination of these costs. Without losing generality, the link travel time is used exclusively in this chapter to represent the link general cost. A path from an origin node (s) to a destination node (g) may be defined as a sequential list of links: (s,j), ...,(i,g). The travel time of the path is the sum of travel times on the individual links. The problem is to find the path that has the minimum total travel time from the origin node to the destination.

This shortest path problem(SPP) has been studied for over thirty years in diverse fields such as computer science and transportation engineering. Due to their computational tractability, most of the research in this area has focused on developing optimal algorithms to solve the problem. The majority of the optimal shortest path algorithms are essentially applications of dynamic programming theory to the search of the shortest path in a graph. The shortest path is found through an recursive decision making procedure from the origin node (or destination node) to the destination node (or origin node).

Most shortest path algorithms have the same standard structure. To describe this procedure, the following notation is introduced. The route cost from the origin node to a particular node i is defined as $L_{(i)}$ and this route cost is commonly referred as the "label" of the node. P is denoted as the list which store the preceding links on the shortest path tree to each node, and $P_{(i)}$ represents the preceding link on the shortest path to node i. Q is denoted as the scan eligible node set which manages the nodes to be examined during the search procedure. The following is the prototype procedure of the shortest path algorithms with the assumption that the algorithm starts from the origin node.

```
Step 1: Initialization:    Set i = o; $L_{(i)} = 0$ ; $L_{(j)} = \infty \ \forall j \neq i$ ; $P_{(i)}$ = NULL.
                           Define the scan eligible node set Q={i};
Step 2: Node Expansion:    Select a node i from Q, and scan each link emanating from
                           node i. For each link a=(i,j)
                           If
                               $L_{(i)} + c_{ij} < L_{(j)}$ ,
                           then
                               $L_{(j)} = L_{(i)} + c_{ij} \ P_{(i)} = a$,
                               Insert node j into Q, and remove a node i from Q;
Step 4: Stopping Rule:     If Q = ∅ then STOP.
otherwise: goto step 2.
```

The major variations between different algorithms pertain to the data structure used to form the *scan eligible node set* and the manner in which the nodes are identified and selected for examination (Gallo *et al.* 1986). Based on the behavior of the algorithm, the optimal shortest path algorithms are usually classified into two categories: label correcting or label setting algorithms (Rilett; 1992).

6.2.1 Label Correcting Algorithm

The label correcting algorithm uses a list structure to manage the scan eligible node set that need to be examined during the shortest path tree building process. It is the variations of the list operation policy that is used to differentiate the label correcting algorithms such as label correcting with queue (Moore, 1969), label correcting with double ended queue (Pape, 1974), and label correcting with threshold lists (Glover *et al.*, 1985).

The major feature of a label correcting algorithm is that it can not provide the shortest path between a root node and another node before the route to every node in the network is identified. The necessity of this type operations (referred to as *one to all* search mode) makes the label correcting algorithms more suitable in situations when many shortest paths from a root node need to be found. Because of this special attribute and the fact that it has historically been found to be the quickest, the label correcting algorithm is often used in these models of most transportation planning applications (Gallo and Pallottino, 1984;Rilett, 1992).

6.2.2 Label Setting Algorithm

In the label setting algorithms, the scan eligible node set is ordered based on the current path cost from the root node, i.e., their labels. During the shortest path search the node with the lowest label is selected for examination and at thesame time, the shortest path to this node is identified. The major difference among the label setting algorithms is the data structure used to maintain the ordered scan eligible node set. Examples include Label setting with sorted list (Dijkastra, 1959), Label setting with binary heap (Tajar, 1983) and Label setting with buckets (Dial, 1969). For convenience in describing the heuristic algorithms in the following sections, the procedure of the label setting algorithm is listed in more detail here:

Step 1: **Initialization:** Set $i = o$; $L_{(i)} = 0$; $L_{(j)} = \infty \ \forall j \neq i$; $P_{(i)} = NULL$.

Define the scan eligible node set $Q=\{i\}$;

Step 2: **Node Selection:** Select and remove the node with the lowest label (travel time) from Q, This is node i;

Step 3: **Node Expansion:** Scan the forward star of node i. For each link a=(i,j)

If

$$L_{(i)} + c_{ij} < L_{(j)} ,$$

then

$$L_{(j)} = L_{(i)} + c_{ij} ; P_{(i)} = a,$$

Insert node j into Q;

Step 4: **Stopping Rule:** If the node i is the destination node, then STOP.

otherwise: goto step 2.

The major feature of a label setting algorithm is that if only a particular route from an origin node to a single destination node is required to be found, the algorithm can be terminated when the label of that destination node is set. This type of operations is usually referred as a *one to one* search mode. As a result, the label setting algorithms are particularly appropriate for the applications such as distributed RGS where the objective is to find the shortest paths between two specific locations.

6.2.3 Computational Performance of the Optimal Shortest Path Algorithms

The computational performances of the label correcting and label setting algorithms have been studied widely from both a theoretical and empirical point in many different research fields. According to the literature which has a focus on transportation networks (Gallo et al., 1984; Hung et al. 1988; Vuren et al., 1988), a number of conclusions pertaining to their characteristics have been identified. Among the label correcting algorithms, the label correcting algorithm with double ended queue and label correcting algorithm with threshold lists are found to be dominant with respect to computational efficiency. The difference in computation time between these two label correcting algorithms is relatively minor for most transportation road network problems and consequently the former has traditionally been used because it is more easily implemented. On the other hand, among label setting algorithms, the label setting algorithm with binary heap is the fastest.

Based on this previous research, the label setting with binary heap algorithm (labeled as LS) and the label correcting with double ended queue algorithm (labeled as LC) are selected as the base algorithms for developing new heuristic shortest path algorithms for one to one search applications.

6.3 HEURISTIC SHORTEST PATH SEARCH METHODS

The optimal shortest path algorithms discussed in section 6.2 tend to be too computationally intensive for real-time one-to-one applications in realistic traffic networks. This "inefficiency" stems from the fact that the algorithms employ "uninformative" outward search techniques without making use of any prior information such as the location of the origin and/or destination nodes. For example, if the origin node was located in the center of the city and the destination node was located in the far south, the optimal search techniques would be just as likely to search for the minimum path routes north of the origin node as they would search south of the origin node. Intuitively, the efficiency of the algorithms could be improved if more information was used in the search process. This latter point was recognized very early by researchers in the AI field and a number of heuristics were proposed that attempted to use various sources of additional knowledge to reduce the search efforts.

The heuristic search strategies can be generally classified into four categories: (i). Limiting the search area, (ii). Decomposing the search problem, (iii). Limiting the search links, and (iv). Some combination of above. In the following sections, these heuristic search strategies and their applications in the shortest path search are explored.

6.3.1 Limiting the Search Area

The non-informative search algorithms, such as optimal label setting and optimal label correcting algorithms, have a fundamental disadvantage in that they examine all the intermediate nodes from origin node to destination node without considering how likely these nodes will be on the shortest path. The idea behind the "limit search area" strategy is to make use of some knowledge about the attributes of the shortest path(s) from the origin node to the destination node to constrain the shortest path search within a certain area. The theory is that the resulting search area would be much smaller than that by a non-informative optimal algorithm. The following sections introduce two methods which implement the "limit search area" strategy: the branch pruning method and the A* algorithm.

6.3.1.1 Branch Pruning Method

The branch pruning method, similar to the well-known branch-and-bound algorithms in operations research, is proposed in this thesis as an attempt to limit the search area by pruning the intermediate nodes that have lower likelihood of being on the shortest path to the destination node.

In a typical urban traffic network represented by a digraph, each link is typically connected only to the neighboring nodes (e.g. intersections) and the travel time on a link is generally correlated with its length. This attribute allows the search area to be constrained within a specified area surrounding the origin node and destination node. The nodes outside this area are assumed to have a lower probability of being on the shortest path and therefore can be dismissed without further examination during the search procedure. The problem is to define the search area such that the computation time can be effectively reduced while at the same time obtaining a good solution. It is proposed in this chapter that a suitable method would be to use the following inequality to bound the search area:

$$L_{(i)} + e_{(i,d)} \leq E_{(o,d)} \qquad (6\text{-}1)$$

Where:

$L_{(i)}$ = the current minimum travel time from origin node o to node i;

$e_{(i,g)}$ = the estimated travel time from node i to destination node g;

$E_{(s,g)}$ = an estimated upper bound of the minimum travel time from the origin node s to the destination node g.

The equation (6-1) can be added to the optimal label setting and label correcting search procedure to form the following heuristic branch pruning shortest path algorithms: the branch pruning label setting algorithm (BP_LS) and the branch pruning label correcting algorithm (BP_LC).

The branch pruning algorithms have exactly the same structure as the optimal algorithms. The only difference is that during the search procedure Equation (6-1) will be tested before a node is selected for examination. For example, the BP_LS algorithm would follow the same steps as the LS algorithm discussed in section 6.2.2, with the exception that the second step would be modified as follows.

Step 2: **Node Selection:** Select and remove the node with the lowest label (travel time) from Q, This is node i;

If $L_{(i)} + e_{(i,g)} > E_{(o,g)}$, then goto step 4.

Basically, the modified step 2 states that if node i is not in the search area it is removed from further consideration. The efficiency of the branch pruning algorithms is schematically illustrated in Figure 6-1. It can be seen that the new heuristic algorithm reduces the search area to that of an ellipse from a circle as examined by an optimal label setting algorithm. In an idealized Euclidean grid network, these heuristic algorithms may be shown to examine a search area that is approximately 9% of the search area used by the LS algorithm (Appendix A).

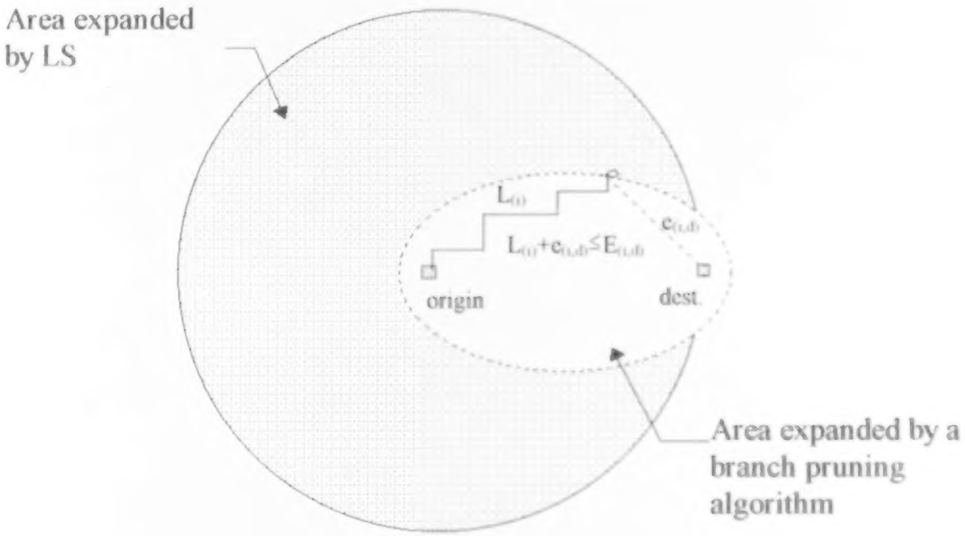

Area expanded by LS
$L_{(i)}$
$e_{(i,d)}$
$L_{(i)}+e_{(i,d)}\le E_{(i,d)}$
origin
dest.
Area expanded by a branch pruning algorithm

Figure 6-1 A schematically illustration of the pruning power of the branch pruning algorithms

It should be noted that the application of the branch pruning method in the LC algorithm is quite unique in the sense that it effectively allows the LC algorithm to be operated in a one-to-one search mode. It can be expected that the BP_LC algorithm would be more efficient as compared to the BP_LS algorithm because they have same search area but the BP_LS has to maintain a ordered list at each search step.

The efficiency and accuracy of the branch pruning shortest path algorithms depend on the quality of the estimation functions of $e_{(i,g)}$ and $E_{(s,g)}$. It should be noted that optimality of the branch pruning shortest path algorithms will be preserved if the estimation $e_{(i,d)}$ is always lower than the minimum travel time from node i to destination node g and at the same time the estimated value of $E_{(s,g)}$ is greater than the minimum travel time from origin node s to destination node g. In addition, the branch pruning shortest path algorithms are the same as their related optimal shortest path algorithms when $e_{(i,g)}$ equals zero and $E_{(s,g)}$ is close to infinite. That is, when the constraint in equation (1) is no longer in effect.

On other hand, if the $e_{(i,g)}$ value is over-estimated and the $E_{(s,g)}$ value is under estimated, then the minimum paths to the destination node may be pruned before the destination node is examined. As a result, the estimation of the value $e_{(i,g)}$ and $E_{(s,g)}$ is critical for implementation of the branch pruning algorithms. There could be many methods by which $e_{(i,g)}$ and $E_{(s,g)}$ can be defined. Because there is a straight dependent relationship between the travel time with the travel distance and travel speed, in this thesis it is suggested that the travel distance $D_{(i,g)}$ and an average travel speed V be used to estimate $e_{(i,g)}$ and $E_{(s,g)}$.

$$e_{(i,g)} = D_{(i,g)} / V \tag{6-2}$$

$$E_{(s,g)} = K\ e_{(s,g)} \tag{6-3}$$

The parameter K, known as the bound factor is used to set the difference between the estimated lower bound on travel time ($e_{(i,g)}$) and the upper bound on travel time ($E_{(i,g)}$).

With these definitions, the performance of the branch pruning algorithms will be controlled by the estimation of travel distance $D_{(i,g)}$, the average speed V and the bound factor K. The travel distance $D_{(i,g)}$ is set to the Euclidean distance from node i to node g and can be directly calculated based on the coordinates of the nodes. The average travel speed (V) is normally related to travel distance and the time of day in an urban environment. For simplicity, it is assumed to be a constant value. With these assumptions, the bound factor K becomes the only controllable parameter in a branch pruning algorithm. The larger the bound factor K, the larger the search area and hence the more likely the optimal solution will be found, and consequently the longer the computing time. A "proper" value of K will be in all likelihood network specific, and quite possibly O-D specific as well. Section 6.4 will demonstrate how this value may be determined using the Edmonton network as an example.

As mentioned previously, one major problem pertaining to the branch pruning shortest path algorithm is that the algorithm may stop without providing any solution if the bound factor used is too small so that all of the branches of the shortest path tree from the origin node are pruned before reaching the destination node. This problem can be resolved by adding a self-adjustment loop in the algorithm, i.e., the algorithm will automatically increase the bound factor K and restart the search from the origin node if a solution has not been found when the scan eligible list is exhausted. Of course, it is desirable to select the value of K such that this does not happen.

Instead of restarting the whole search, an alternative method is to record all of the pruned nodes during the search procedure and put them back into the scan eligible node list once it is exhausted. It would be expected that the latter approach would be faster than the former method. The reason is that at each iteration the latter approach will use the search results of the previous iteration instead of restarting the search from scratch again as the former method does. The disadvantage of using the second method is that it would require extra memory to store information on all of the pruned nodes.

The K value at each iteration can be determined either by simply increasing a prespecified proportion or by using the information in the previous iteration. In the latter method, the K value can be determined by using the ratio of maximum value of the estimated route travel times going through the pruned nodes to the estimated lower bound on the route travel time from the origin node to the destination node:

$$K_{n+1} = \frac{\max_{i \in P_n}\{ L_{(i)} + e_{(i,g)} \}}{e_{(s,g)}} \tag{6-4}$$

Where P_n is the pruned node set at iteration n.

It should be noted in all of the above analysis that a good initial K value can avoid either looping for the self adjustment or searching an unnecessary large area.

6.3.1.2 A Algorithm*

The A* algorithm was first proposed by Hart (1968) and further extended by Nilsson (1971), Pohl (1971) and Pearl (1984). The strategy behind this algorithm is to change the order in which nodes are examined such that the nodes that have higher "likelihood" of being on the minimum path are given priority over those with a lower "liklihood". The performance of the A* algorithm has been well studied with respect to Euclidean networks where the objective is to minimize the travel distance. An empirical study has shown that the A* algorithm examines less than about 10% of the nodes that would be examined by an optimal label setting algorithm (Golden *et al.*, 1978). It was also shown that (Sedgewick, 1986) the A* finds the shortest path in many Euclidean graphs with an average computation effort O(n), compared to O(n.log n) required for the LS algorithm. This section will focus on how this algorithm can be implemented in traffic networks.

The A* algorithm makes use of an additive heuristic evaluation function $F_{(i)} = L_{(i)} + e_{(i,g)}$ for node i, where $L_{(i)}$ is the travel time of current evaluated path from origin node to node i and $e_{(i,g)}$ is an estimated travel time for node i to destination node g. The sum of these two functions, $F_{(i)}$, is the merit of node i, representing how likely node i will be on the shortest path. The lower the merit of a node, the more likely the shortest path will go through this node. Based on this idea, the algorithm does a best first search, i.e., it maintains an ordered list of nodes to be scanned according their merits and selects a node whose merit is the lowest among all the nodes for expansion. The selected node is expanded by scanning its forward star (or backward star), evaluating them according to their F value, and inserting them into the ordered scan eligible node set. This continues until the destination node is chosen for expansion. The A* algorithm has a similar structure as the LS algorithm except that the evaluation function $F_{(i)}$ would be used as the label for determining the node examination order. The main difference lies in step 3 which is modified as shown below:

```
*Step 3*: **Node Expansion:**   Scan the forward star of node i. For each link (i,j)
                                If
                                    $L_{(i)} + c_{ij} + e_{(j,g)} < F_{(j)}$ ,
                                then
                                    $L_{(j)} = L_{(i)} + c_{ij}$ ; $F_{(j)} = L_{(i)} + c_{ij} + e_{(i,g)}$ ; $P_{(i)} = a$,
                                    Insert node j into Q;
```

Because the A* algorithm proceeds as a best first search, the node which satisfies the following inequality would be examined before the algorithm terminates (the destination node is examined):

$$L_{(i)} + e_{(i,g)} \leq L_{(g)} \tag{6-5}$$

This attribute makes the A* algorithm hold an important property, that is, it is guaranteed to find the optimal solution as long as the heuristic function never overestimates the actual travel time (Nilsson, 1971).

Equation (6-5) also defines the area which would be examined during the shortest path search. The resulting search area becomes elliptical in shape rather than the circular shape associated with the LS algorithm. The pruning power of an A* algorithm is schematically illustrated in Figure 6-2. It should be noted that all the nodes on the edge of the search area (ellipse) will need to be examined during the search procedure. This contrasts with the situation of the branch pruning algorithm (Figure 6-1) where some nodes on the edge are no longer in the scan eligible node list after proving to be located out of the assumed feasible solution area. Therefore the branch pruning algorithm is theoretically faster as it can maintain a smaller scan eligible node set as compared to the A* algorithm.

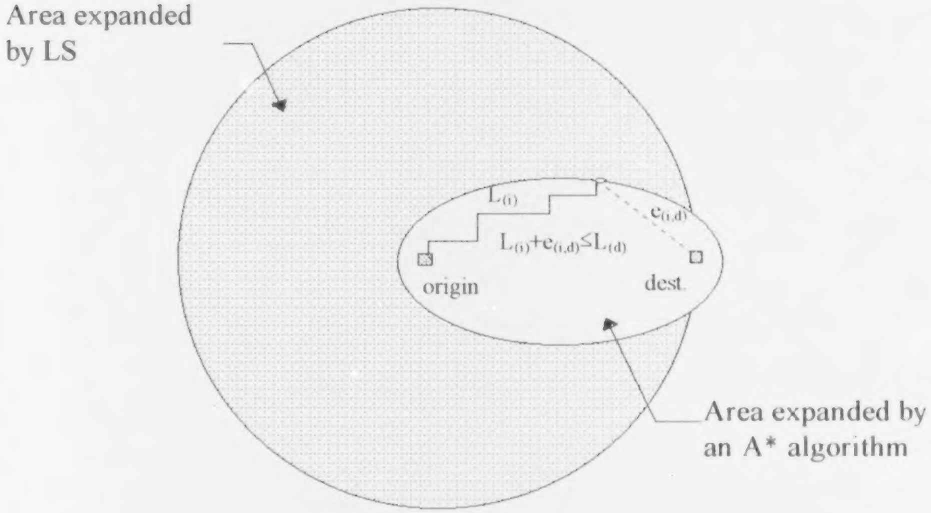

Area expanded by LS
$L_{(i)}$
$e_{(i,d)}$
$L_{(i)}+e_{(i,d)}\leq L_{(d)}$
origin
dest.
Area expanded by an A* algorithm

Figure 6-2 A schematically illustration of the pruning power of the A* algorithms

The performance of the A* algorithm depends on the quality of the heuristic function used. Similar to the branch pruning method, $e_{(i,g)}$ can also be estimated by using equation (6-2). When the travel distance is estimated by using Euclidean distance, the average speed V becomes the only controllable parameter in this algorithm. In order to keep $e_{(i,g)}$ as a lower bound estimate, an upper bound average speed should be used. However, it should noted that the higher the average speed used, the larger the search area will be, and thus the more computational effort required. As the value of V approaches infinity, the estimated travel time approaches zero and the A* algorithm becomes the same as the LS algorithm. On other hand, the use of a smaller value of V can limit the search area more significantly and the algorithm is faster, however the probability of finding the optimal solution is reduced.

6.3.2 Decomposing the Search Problem

It has been well recognized that the computational effort required to solve the shortest path problem increases exponentially as the distance between the origin node and the destination node increases (Korf, 1986). As a result, if an original problem can be decomposed into smaller sub-problems, the computational saving can be realized. This section introduces how this strategy is implemented in the bi-directional search method and the subgoal method.

6.3.2.1 Bi-directional Search Method

Most traditional search methods are uni-directional in the sense that they seek the problem solutions from the initial stage to the goal stage. A bi-directional search method, first proposed in the 1960's (Nicholson, 1966), attempts to divide the search procedure (problem) into two separate procedures (problems). One search proceeds forward from the initial stage while another search processes backward from the goal stage. The solution is identified when these two search procedures meet at some middle stage(s).

This concept is well suited for solving the shortest path problem. The algorithm simultaneously builds the shortest path trees forward from the origin and backward from the destination until some stopping criteria is met. Figure 6-3 schematically illustrates the concept and effectiveness of this method as compared to the LS algorithm.

The stopping criteria that guarantees to find the shortest path is given below : (Nicholson, 1966)

$$L^{o}_{(i)} + L^{d}_{(i)} \leq \min_{j\in N}\{L^{o}_{(j)}\} + \min_{j\in N}\{L^{d}_{(j)}\} \tag{6-6}$$

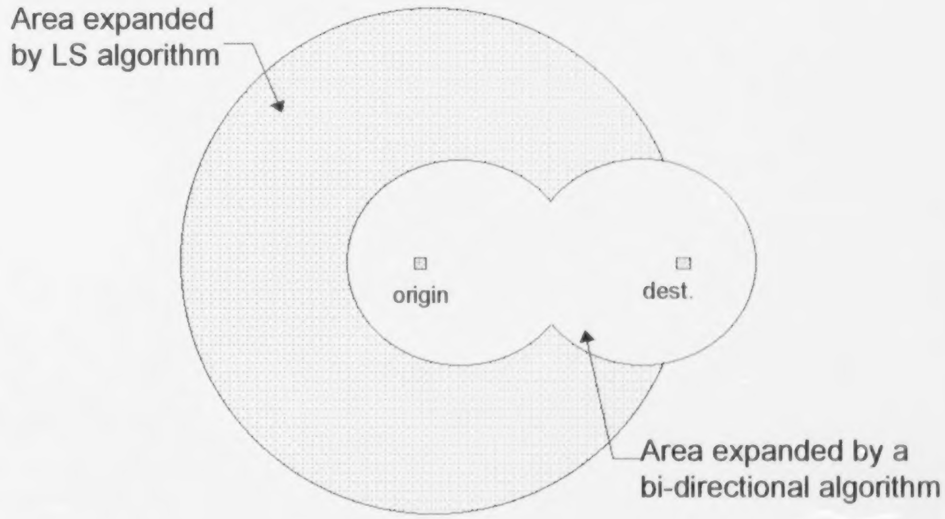

Area expanded
by LS algorithm
origin
dest.
Area expanded by a
bi-directional algorithm

Figure 6-3 A schematically illustration of the pruning power of the bi-directional search algorithms

Where the label $L_{(i)}$ with the superscript o and d represent respectively the label of node i on the shortest path tree out of the origin node (o) and the label of node i on the shortest path tree out of the destination node (d). With this correct stopping criteria, the bi-directional search algorithm has been found inferior to the uni-directional algorithms (Dreyfus,1969). In this thesis, this bi-directional searching algorithm was modified by introducing a new stopping rule. The new stopping rule is defined as that the tree building procedure from both ends will not stop until given numbers (say m) of candidate paths are found.

Two steps are involved in the new bi-directional shortest path search method. The first step is to generate a given number of candidate paths by using algorithms such as the LS algorithm or A* algorithm to build shortest path trees from both ends simultaneously. Whenever there is a node whose labels on both shortest path trees are set, the corresponding path, which goes though this node from the origin node to the destination node, is then recorded as a candidate path. This procedure will not stop until it generates m candidate paths. The second step selects the best path among those candidate paths.

With the new stopping rule , heuristic bi-directional shortest path algorithms can be created by combining the bi-directional search method with the LS algorithm and the other heuristic methods. For example, the bi-directional search method can be combined with the branch pruning method and label setting algorithm to form the heuristic bi-directional branch pruning label setting algorithm (B_BP_LSm), and it can be combined with the A* algorithm to form the heuristic bi-directional A* algorithm (B_A*m).

It should be noted that although the resulting algorithms would be faster with the introduction of the new stopping rule, they are not guaranteed to find the optimal solution. The performance of a heuristic bi-directional algorithm is controlled by the parameter m used in the algorithm. The larger the value of m, the more likely the optimal solution will be found and the longer the computation time will be. By selecting an appropriate value of m, a satisfactory balance between accuracy and efficiency can be reached.

The performance of the bi-directional search algorithms may also be influenced by the method used to alternate between the forward search and the backward search. Although iterating equally between both searches would be the simplest method, it is not the most efficient. The best strategy involves identifying the minimum path search that has the fewest nodes which have been examined but have not had the minimum path identified. That is, the computational effort is concentrated on the search having the least nodes to examine and sort during each iteration. Intuitively, the search that is in a sparse area of a network will get priority.

Finally, it should be noted that the bi-directional algorithms cannot be directly used to find the shortest path in a dynamic network. The reason is that in a dynamic network the shortest path tree from the destination node is not unique but rather depends on the arrival time at the destination node which is not known in practical situations (usually, only the departure time at the origin node is known).

6.3.2.2 Subgoal Method

A subgoal is defined as an intermediate state which is part of the optimal solution of a problem. For the shortest path search problem in a road traffic network, subgoals indicate the locations (nodes) where the shortest path from an origin to a destination will go through. With advance knowledge of the subgoal node(s), the problem of finding the shortest path from the origin node to destination node can be decomposed into two or more smaller problems. For example, if there is one subgoal node, the original problem can be solved by solving two sub-problems: one is to find the shortest path from the origin node to the subgoal node while another is to find the shortest path from the subgoal node to the destination node.

The efficiency of the subgoal method depends on the number and location of the subgoal nodes. The greater the number of the subgoal nodes and the more uniformly they are distributed between the origin node and the destination node, the more the computational saving will be. In an idealized situation, M subgoal nodes equally located between the origin node and the destination node in uniform and infinite network, the reduction of the search area using the subgoal nodes would be $^1/_{(M+1)}$. For example, if a single subgoal node is known and located around the middle of the shortest path from an origin to a destination, then the use of this subgoal node will reduce the search area approximately 50% as compared to the LS algorithm as shown in Figure 6-4. Of course, this method can be used in conjunction with some other techniques discussed in previous sections to further reduce the search area.

Although the computation efficiency of the subgoal method is obvious, the penalty of this search reduction is that the solution may not be optimal, i.e., the path going through the subgoal node may not be a real shortest path. Its application depends on whether or not such valuable information is available. In a road network routing environment, potential subgoals include bridges, intermediate stops and driver's preference.

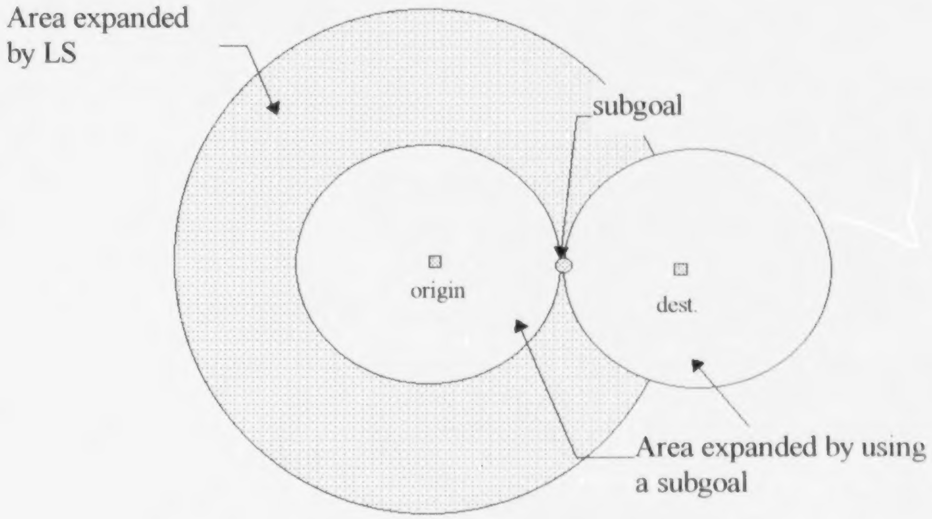


Figure 6-4 Pruning power of using a sub-goal

6.3.3 Limiting the Search Links

As seen in section 6.2, during the shortest path search procedure the main decision variable is identifying the best link emanating from each node. In a traditional shortest path algorithm, when a node is selected for expansion, each link from this node will be equally examined without considering how likely it will be on the shortest path(s) or will be used as part of the optimal path in practical situations. The idea of limiting search links is to systematically skip the examination of the links that have very a low probability of being on the shortest path in practical situations. This idea can be effectively implemented by using the hierarchical search method discussed in the following section.

6.3.3.1 Hierarchical Search Method

The hierarchical search is well known in the AI field is also known as an 'abstraction' problem solving strategy. It was first presented by Plolya et al. (1945) and further explored by Sacerdoti (1974) and Korf (1987). The basic idea behind the hierarchical search is that in order to effectively find a solution of a complex problem, the search procedure should at first concentrate on the essential features of the problem without considering the low level details, and then complete the details later. This method can be explained by the procedure that a driver manually finds a route between two locations on a map. Typically, the driver first examines the major roads in the area adjacent to the origin location and destination location, and then finds the access roads to the major road from both origin and destination locations. This technique was proved very effective in reducing the complexity of large problems, and is also the only one which has potential to beat the exponential increase of computation time in terms of travel distance (Korf, 1987). The computational saving in an idealized two level hierarchical network is shown in Appendix B.

Due to the hierarchical topological structure inherent to urban traffic networks, hierarchical search techniques can be effectively used to find shortest paths in these networks. Most transportation road networks are designed as a combination of different functions of roads to provide services to different types of trips. For example, freeways and arterials mainly service the long distance trips while local roads or collectors are used to access the arterial or for local trips. Therefore as an optimal route it should also meet the road functional planning principles. For instance, a long trip should not take a path going through a lower class of roads such as residential roads even if it has a lower travel time.

The algorithmic implementation of the hierarchical search method for pathfinding takes an opposite procedure as that described above. This can be illustrated by using simple two level hierarchical networks as shown in Figure 6-5. A bi-directional search algorithm can be used to effectively carry out the hierarchical search strategy. The search starts from the lower level network where the origin node and destination node are located, and builds shortest path trees forward from the origin node and backward out of the destination node on that level network. This search at the lower level network will be bounded by the links (or nodes) which also reside in the higher level network. Once the search climbs on the higher level network, that is, when all the nodes in the scan eligible node set also belong to the higher level network, it will proceed using a bi-directional search algorithm within that level until the shortest path is found.

There are two problems to be solved before this search strategy can be effectively implemented. The first problem is to identify the methodology for disaggregating the network into different levels and subsequently how to represent the hierarchical networks. For the application in a road traffic network, the first part of the problem may be trivial because the functional planning and the classification of the road network can be readily used for abstraction.

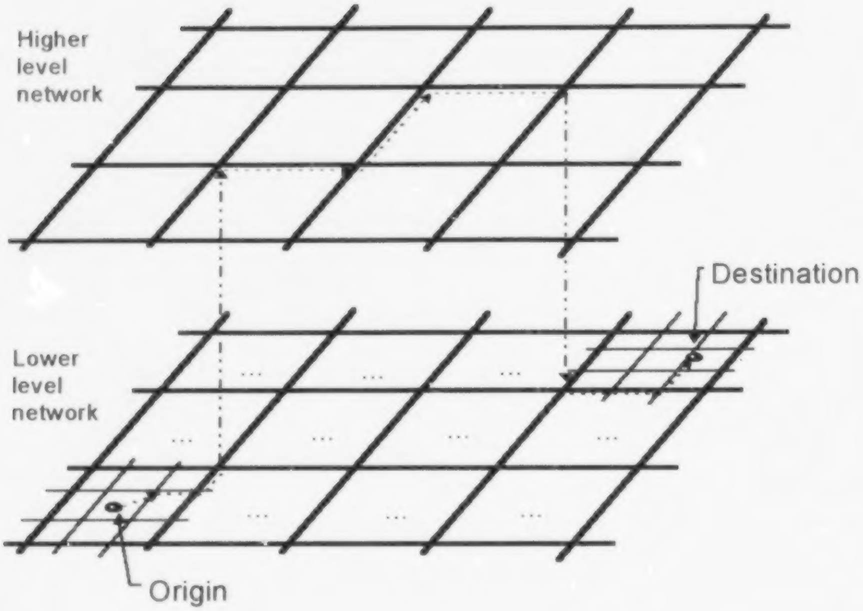


Figure 6-5 A schematically illustration of the shortest path search procedure in a two level hierarchical network

The method of representing and managing the hierarchical networks could be critical. In principle, the lower level should contain all data of the levels above it. The relations between neighboring levels can be established through dummy links between intersections in the higher level network with same intersections in the lower level network. These dummy links provide free travel between the different levels. It should be noted that this representation requires redundant storage about the network data.

The second problem is to establish the rule by which the search is controlled, i.e., when the search should "go up" to a higher level network or "turn down" to a lower level network. As discussed above, the bi-directional search method can effectively be used to search the shortest path in hierarchical networks. The control used is the search area bounded by the higher level networks (nodes and links). Consequently, this control method will not allow any shortcuts such as moving from one arterial to another by going through a residential road. In a traffic network there are certainly many types of shortcuts existing and some of them are even unavoidable. For example, it is usually necessary to shift between two parallel freeways by going through an arterial which connects them. In this situation, the search procedure discussed above will miss the actual shortest path (the path which has a shortcut). To avoid this problem, one method is to increase the search depth at each level. For example, instead of being bounded by the higher level network, the search can be expanded one block further at each level. Another method to achieve this is to use the number of nodes searched at each level as a control (Kuznetsov, 1993).

6.4 COMPUTATIONAL STUDY

The section 6.3 has provided a in-depth discussion on various heuristic search strategies and their applications in shortest path searches. The qualitative analyses, although providing some interesting insight on the performance of a heuristic, are not sufficient to assess a heuristic algorithm for practical applications. For example, the following questions need to be answered:

a) How much improvement can a heuristic algorithm offer compared to an optimal algorithm when applied in a realistic traffic network?

b) What parameter value is best for the heuristic algorithms? and

c) Which is the best heuristic algorithm for a given application such as RGS and AVDS?

In this section, a computational study is conducted as an attempt to answer some of these questions. The algorithms to be investigated are summarized in Table 6-1. It should be noted that the sub-goal method and hierarchical search method discussed in previous section are not covered in the following computational study because of the time limitation for this thesis study.

Table 6-1 Heuristic algorithms and their acronyms

ACRONYM	ALGORITHM
LS	Label setting algorithm with binary heap
LC	Label correcting algorithm with double end queue
BP_LS	Label setting combined with branch pruning method
BP_LC	Label correcting combined with branch pruning method
B_BP_LS	Bi-directional label setting with a correct stopping rule combined with branch pruning method
B_BP_LSM	Bi-directional label setting with the new introduced stopping rule combined with branch pruning method
A*	A* algorithm with binary heap
B_A*	Bi-directional A* with correct stopping rule
B_A*M	Bi-directional A* with new introduced stopping rule

Before assessing the performance of these heuristic algorithms, there are three issues that need to be addressed. The first issue is to identify the appropriate method for defining the test network on which the heuristic algorithms will be used . In our study, a single real world road network is used as a test network instead of many randomly generated graphs as most experimental studies have done. The reason for this is that the research focuses on developing heuristic shortest path algorithms for RGS and AVDS and the application environments of these systems are urban traffic networks that usually have a similar network structure. The same network described in Chapter 4 is used in this study. It should be noted that the primary objective of this study is to show relative performance of the algorithms instead of arriving at a definite answer.

The second issue is how to select the evaluation criteria used to measure the quality of solution and computational effort of an algorithm. In this study, CPU time and relative CPU time saving are used to measure the computation effort of an algorithm while the relative solution error is used as a measure of the quality of a solution. All the relative performances are compared to the LS algorithm without further notation.

The final issue is the implementation environment including software and hardware. All the programs in this study are coded with C++ and executed under Microsoft Windows on a 486 compatible computer with a 50 MHz speed and 8 MB RAM.

The overall methodology for the analysis is as follows. O-D pairs are randomly generated and their shortest paths are found by using each heuristic algorithm. The CPU times and the estimated route travel times for each algorithm are recorded for analysis.

The analysis concentrates on two aspects. The first one is the behavior of each heuristic shortest path algorithm in terms of their sensitivity to the parameters used and search depth (or O-D travel time). The second aspect is the relative performance of all the heuristic shortest path algorithms for different applications. The following sections summarize these results.

6.4.1 Performance of Branch Pruning Algorithms

This section will discuss the empirical performance of the branch pruning algorithms including the algorithms BP_LC, BP_LS, B_BP_LS and B_BP_LSm. The analysis will first focus on the relationship between their performance with their common parameter, i.e., the bound factor K used in them. The m value in the B_BP_LSm algorithm is set to 1. Figure 6-6 illustrates the relationship between the CPU time saving of each algorithm as a function of the K value. The BP_LC algorithm is the most efficient among these tested with a CPU time saving of approximately 40%. The bi-directional algorithm with a correct stopping rule (B_BP_LS) is always inferior to its respective uni-directional algorithm (BP_LS). This result confirms the argument presented in previous research (e.g., Drefus, 1969). By introducing the new stopping rule as described in Section 6.3., the CPU time saving for B_BP_LSm are about 30%. It can also be seen that there is an optimal range of K values (approximately 1.4~1.8 for the Edmonton network) which gives the highest CPU time saving for each algorithm. This is a result of the fact that more self adjusting loops are required when the K value used is relatively small while the search area becomes unnecessarily large if the K value is relatively high. In both cases computation benefits of the algorithm are lost to the resulting increased travel time.

Although the branch pruning algorithms are superior to the optimal label setting algorithms (LS) in terms of computation efficiency, they are not guaranteed to find the optimal solution. Figure 6-7 shows a graph of the relative error of each algorithm as a function of the bound factor K. It may be seen that the maximum relative error of the B_BP_LSm algorithm is relatively high compared to the other three algorithms (1.25% vs. 0.1%); however, it decreases quickly as the K value increases. For example, when the value of K is increased to 1.8, the B_BP_LSm algorithm has an approximately 0.2% relative error. This remaining error is mainly caused by the introduction of the new stopping rule. It may be expected that this error would be reduced by increasing the value of m used in the B_BP_LSm algorithm. This will be further discussed in section 6.4.3

From the analysis of Figure 6-6 and Figure 6-7, it may be concluded that for the Edmonton network, K values of 1.4~1.8 are appropriate for the BP_LC algorithm. The K values would result in an approximately 50% decrease in computation time as compared to the LS algorithm with a corresponding relative error of approximately 0.1%. The B_BP_LSm algorithm appears to perform adequately at a K value of 1.8, and this results in a 30% CPU time saving and a relative error of 0.1% as compared to the LS algorithm.

In one to one search mode, the actual performances of these algorithms are also dependent on how far an O-D pair is apart. Figure 6-8 illustrates the relationship of the CPU time consumed for the LS, BP_LS, BP_LC and B_BP_LSm algorithms to find the shortest path of an O-D pair as a function of the O-D travel time. Each point in the graph represents about the average results of 100 randomly generated O-D pairs. As would be expected, as the O-D travel time increases so does the calculation time required to identify the shortest path, and the algorithm LS has the highest increase rate. Once again, the BP_LC algorithm show its significant domination over other algorithms in terms of the CPU time increase rate with respect to the O-D travel time.

6.4.2 Performance of A* Type Algorithms

This section examines the empirical performance of the A* family of algorithms including the A*, B_A* and B_A*m algorithms. The analysis will first focus on the relationship between their performance with their common parameter, i.e., the average speed V used in them. The m value in B_A*m is set to 1. Figure 6-9 shows a graph of the CPU time saving as a function of the average speed V. As would be expected each A* type algorithm provided a significant computational saving compared to the LS algorithm. The CPU time saving of each algorithm decreases as the average speed increases. This is simply because the increase of the parameter V increases the search area. Similar to the results in the last section, the bi-directional A* algorithm B_A* is not faster than its respective uni-directional algorithm A*. With the new stopping rule discussed in Section 6.3, the B_A*m algorithm is found to be significantly faster than the B_A* algorithm as evidenced by the CPU time saving of about 40%~65% compared to the LS.

It can also be seen in Figure 6-9 that the CPU time of algorithms A* and B_A* decrease approximately twice as fast as the B_A*m as the average speed V increases. This pattern would be expected as the B_A*m algorithm, on average, expands about half of the search area of the A* or B_A* algorithms, and the search area of a A* type algorithm is proportional to the average speed.

The solution quality of the A* type algorithms is shown in Figure 6-10. It can be found that while the maximum relative errors of these algorithms are rather high (13%), they decrease quickly as the average speed increases. When the value of V is over 80 km/h, the A* and B_A* algorithms have almost no estimation error while the B_A*m algorithm has an approximately 0.5% relative error. This remaining error is mainly caused by the introduction of the new stopping rule instead of the attribute of the A* type algorithms. This will be further discussed in the next section.

Figure 6-11 illustrates the relationship of the CPU time for each algorithm to calculate the shortest path of an O-D pair as a function of the O-D travel time. It may be seen that the CPU time for the A* algorithm increases much faster than the B_A*m algorithm. It can therefore be expected that for algorithm B_A*m the average CPU time saving would be greater in the case of larger traffic networks.

From the empirical analysis as shown in Figure 6-9 and Figure 6-10, it may be concluded that for the Edmonton network, an average speed of 70 km/h is appropriate for the A* algorithm. This would result in an about 40% decrease in computation time as compared to the LS algorithm and 0.25% relative error. The B_A*m algorithm performs adequately at a V value of 80 km/h with a 50% CPU time saving and a relative error of 1% as compared to the LS algorithm.

6.4.3 Performance of Bi-directional Search Algorithms

This section will focus on the empirical performance of the bi-directional algorithms with respect to the parameter m. Figure 6-12 shows a graph of the CPU time saving of the B_A*m algorithm and the B_BP_LSm algorithm as a function of the parameter m. The parameter average speed V is set to 60 km/h in both algorithms. From Figure 6-12 it may be seen that the CPU time saving of both algorithms decreases as the value of m increases and the rate of decrease is relatively constant. Compared to the B_BP_LSm algorithm, the B_A*m algorithm is approximately 20% lower in CPU time saving and 1% higher in relative error. For both algorithms, the increase of the m value from 1 to 10 results in an approximately 1% decrease in the relative error.

The relative error of B_BP_LSm algorithm decreases quickly to a stable value (about 0.25%) when the m value increases from 1 to 4 while the corresponding decrease of CPU time saving is about 8%. This may indicate that an m value of 4 is appropriate for the B_BP_LSm if applied in the Edmonton network. For the B_A*m algorithm, because there is no significant change of the rates in both CPU time saving and relative error, selection of the m value could depend on the applications (e.g., how much relative error is acceptable).

6.4.4 Selection of Heuristic Algorithms

It is difficult to select the best one among all the heuristic algorithms because each heuristic algorithm usually uses specific parameter(s) in them and the evaluation criteria are usually conflicting (e.g., the lower computation time is usually accompanied by higher solution error). Therefore, the selection of the best heuristic algorithm is really conditional to the application requirements, that is, how quickly a solution is required and how much calculation error the application can tolerate. This section attempts to quantify all of the heuristic algorithms using the same evaluation criteria and examine their relative performances.

Figure 6-14 shows the location of each heuristic algorithm in the evaluation metric based on two measures: CPU time and relative error. It can be seen that the heuristic algorithms generally provide 20% to 60% computational time savings as compared to the optimal algorithm (LS). The average estimation error is less than 0.5% with respect to an average travel time of 1939 seconds. The performance of the branch pruning label correcting algorithm (BP_LC) is comparatively good with a small relative estimation error (0.01%) and a significantly high CPU time saving (40%~60%).

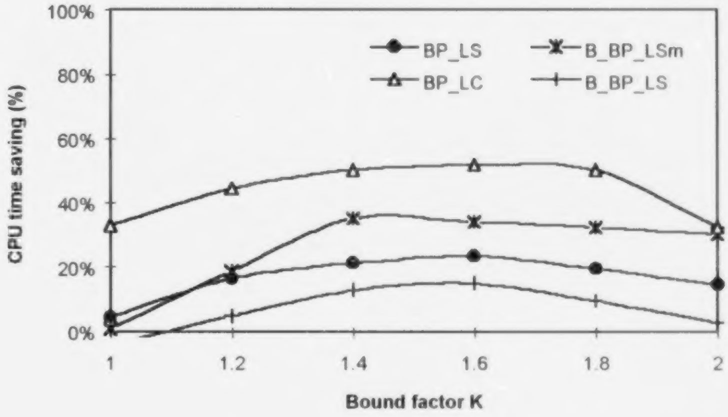


Figure 6-6 Computational efficiency of the branch pruning algorithms

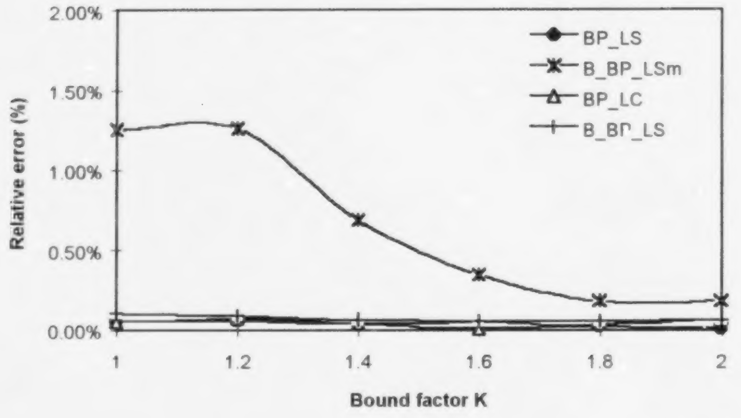


Figure 6-7 Solution quality of the branch pruning algorithms

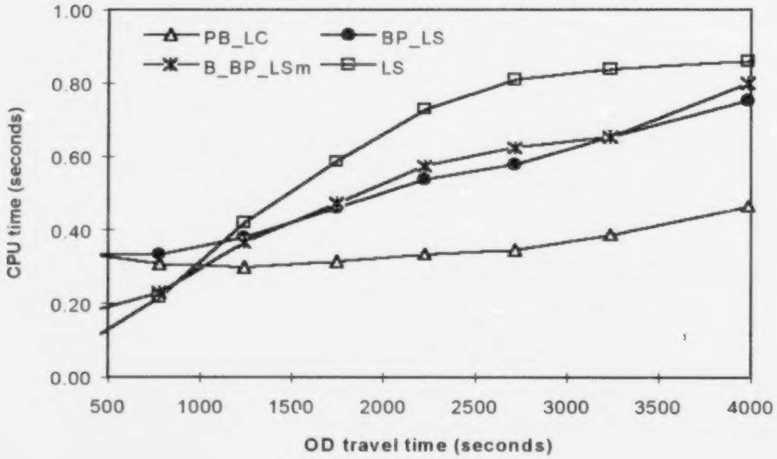


Figure 6-8 CPU time vs. O-D travel time: branch pruning algorithms

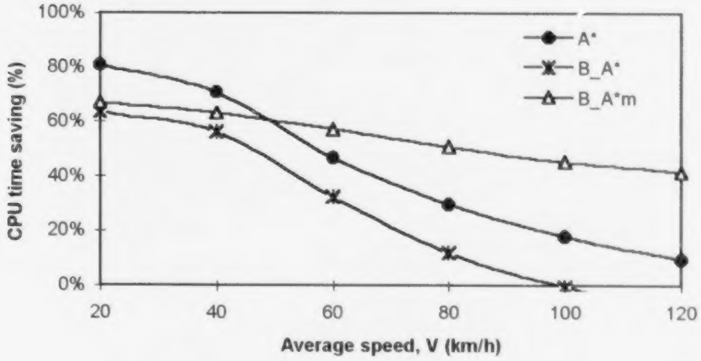


Figure 6-9 Computational efficiency of the A* type algorithms

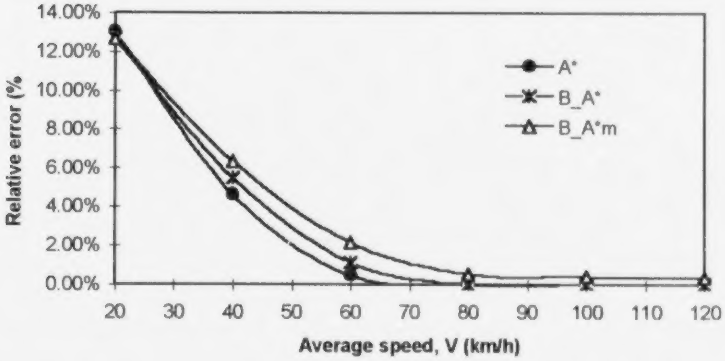


Figure 6-10 Solution quality of the A* type algorithms

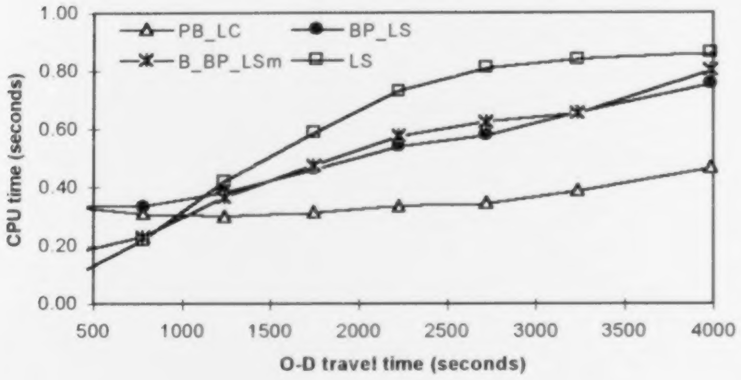


Figure 6-11 CPU time vs. O-D travel time: A* type algorithms

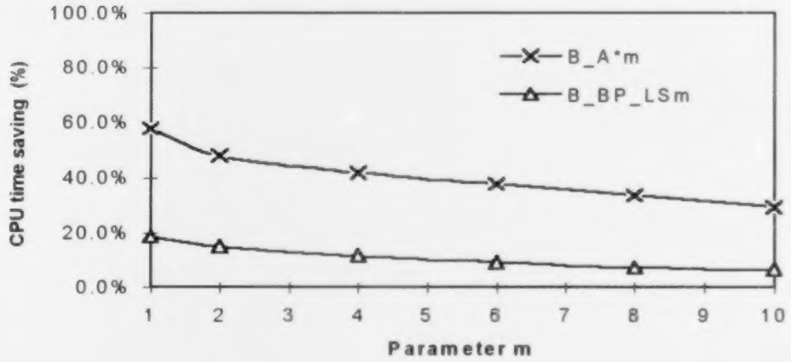


Figure 6-12 CPU time vs. parameter m: bi-directional algorithms

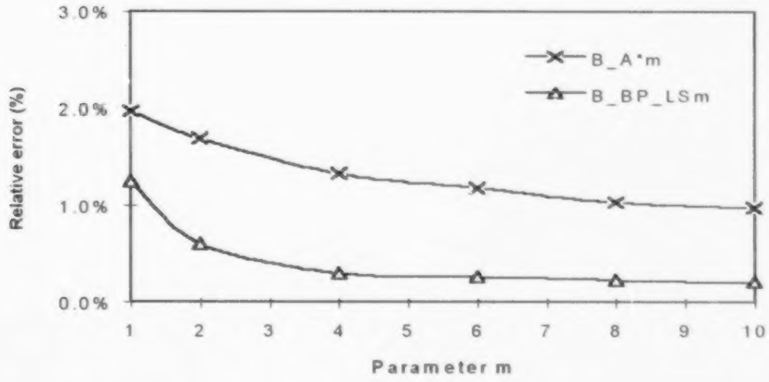


Figure 6-13 Relative error vs. parameter m: bi-directional algorithms

The heuristic bi-directional A* algorithm (B_A*) was found to be slightly more efficient as compared to the BP_LC algorithm, however the former has a significantly higher estimation error (0.5%). From Figure 5 it can also be noticed that the algorithms B_A*m, BP_LC, A* and LS/LC form a dominating core. That all the other algorithms are always inferior in terms of both solution quality and computational efficiency.

It should be noted that this metric is based on their empirical performance applied in the Edmonton network, therefore their relative position could be shifted if a different size of network is used. For example, although the BP_LC algorithm dominate over the B_A*m algorithm with respect to speed and error, the domination may change if a larger network is used. The performance of the heuristic algorithms are also sensitive to the parameters used in them, as shown in the previous sections.

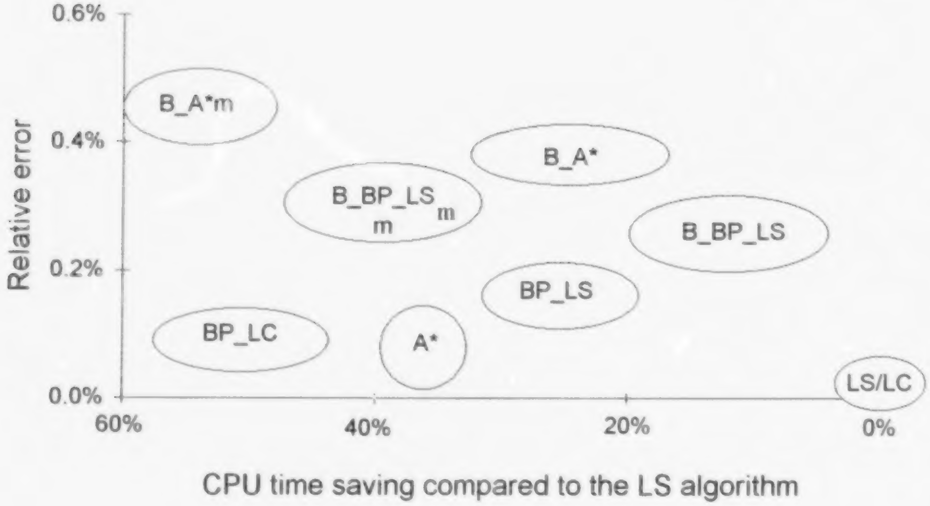


Figure 6-14 Relative performance of the heuristic algorithms

6.5 CONCLUSIONS

This chapter has demonstrated how various heuristic search methods from the AI field can be efficiently applied in finding the shortest path in an urban traffic network. Several heuristic shortest algorithms have been developed and their empirical performance has been studied by using a real size network. The following conclusions are summarized:

1. On various heuristic shortest path algorithms

    - The branch pruning method proposed in this research uses the information on the estimated maximum travel time from the origin node to destination node and the estimated lower bound of travel time between any pair of nodes to bound the solution search area. It was found that this method can be readily incorporated into the shortest path finding algorithms. The resulting heuristic algorithms include parameters and are therefore easily customized to meet the requirements of both computational efficiency and solution quality;
    - The traditional A* algorithm can be modified to take into account the travel speed effect in the travel time estimation (evaluation function) when it is applied to find the shortest path in a traffic network. With a parameter included, the A* algorithms can be adjusted to reach a trade-off between computational efficiency and solution quality;
    - In order to overcome the inefficiency of the traditional bi-directional searching algorithm, a modified bi-directional searching method was proposed by introducing a new stopping criteria in the searching procedure. The new algorithm has been shown to be very effective in finding the shortest path in a traffic network;
    - The heuristic search methods can be combined with each other to generate more powerful algorithms.

2. On the relative performance of the heuristic shortest path algorithms

- Among all the heuristic algorithms the branch pruning label correcting algorithm (BP_LC) generally gave the best results in terms of both computation efficiency and solution quality. Although not guaranteed to find the optimal routes, its relative error in route travel time was relatively small (less than 0.1%) while its computation saving was significant (30~60%);

- Although the algorithm B_A*m is slightly faster than the BP_LC algorithm but has the shortcoming of a higher estimation error (0.25%). It has an important advantage of being less sensitive to the route time for long trips. This attribute could make it more favored than BP_LC in the case that the underlying network is very large (compared to the Edmonton network);

- With an appropriate value of the parameter V, the A* algorithm can be guaranteed to provide optimal solutions while still reducing computation time (10%~30% CPU time saving compared to LS). This is in contrast to that of all other heuristic algorithms;

- In the situation where the link travel times in the network are dynamic and thus the bi-directional algorithms can't be directly used, the BP_LC and A* algorithm would be the best selections;

- Finally, it should be emphasized that the empirical performances of the heuristic algorithms pertain to the test network, the computing platform and the coding language that are used in this study. However it could be expected that the relative saving of computation time would be higher if these heuristic algorithms are applied in larger networks.

REFERENCES:

Dial, B. R, (1969), "Algorithm 360:Shortest Path Forest With Topological Ordering" *Communs Ass. Computing. Mach.* **12**, pp. 632~633.

Dijsktra, E. (1959), "A Note on two Problems in Connection with Graphs," *Numerical Mathematics* **1** , pp. 269~271.

Dreyfus, E. S., (1969), "An Appraisal of Some Shortest Path Algorithms," *Operations Research* **17**, pp. 395~412.

Gallo, G. and S. Pallottino, (1984), "Shortest Path Methods In Transportation Models," In *Transportation Planning models* by M. Florian (editor), pp. 227~256.

Gallo, G. and S. Pallottino, (1986), "Shortest Path Methods: A Unifying Approach," *Mathematics Program Study* 26, pp. 38~64.

Glover, F., D. Klinkgman and N. Philips, (1985), "A new Polynomial bounded Shortest Path Algorithm," *Operations Research*, Vol. **33**, pp. 65~73.

Golden, L. B. and M. Ball, (1978), "Shortest Paths with Euclidean Distance: An Explanatory Model" *Networks*, Vol. 8, pp. 297~314.

Guzolek, J. and E. Koch, (1989), "Real-time Route Planning in Road Network" *VINS*, September 11-13, Toronto, Ontario, Canada, pp. 165~169.

Hart, E. P., N. J. Nilsson, and B. Raphael, (1968), " A Formal Basis for the Heuristic Determination of Minimum Cost Paths," *IEEE Trans. System Science and Cybernetics*, Vol. SSC-4, No. **2**, pp. 100-107.

Hung, S. M. and J. J. Divoky, (1988), "A Computational Study of Efficient Shortest Path Algorithms," *Comput. Operations Research* Vol. **15**, No. 6, pp. 567~576.

Korf, R. E., (1987), "Planning as search: A Quantitative approach" *Artificial Intelligence*.

Kuznetsov, T., (1993), "High Performance Routing for IVHS," *IVHS America 3rd Annual Meeting*, Washington, D. C. April, 1993.

Moore, E. F, (1959), "The Shortest Path Through A Maze" In Proceedings of the International Symposium on Theory of Switching, Harvard University Press, Cambridge, Mass., pp. 285~292.

Newell, A. and H. A. Simon, *Human Problem Solving*, Prentice-Hall, Englewood Cliffs, N. J., 1972

Nicholson, J. A. T., (1966), "Finding the Shortest Route Between Two Points in a Network," *Computer J.* **9**, pp. 275~280.

Nilsson, J. N., *Problem-Solving Methods in Artificial Intelligence*, New York, McGraw-Hill., 1971.

Pape, U. (1974), "Implementation and Efficiency of Moore Algorithms for The shortest Route Problem," *Mathematical Programming* 7, pp. 212~222.

Pearl, J., *Heuristics Intelligent Search Strategies for Computer Problem Solving*, Addison-Wesley Publishing Company, 1984.

Pohl, I., (1971), "Bi-directional Search", *Machine Intelligence* 6, pp. 127~140.

Rilett, L. R. (1992), "Modeling of TravTek's Dynamic Route Guidance Logic Using the Integration Model" Ph.D. Dissertation, Queen's University.

Rilett, L. R. , C. Blumentritt and L. Fu, (1994), "Minimum Path Algorithms for In-vehicle Route Guidance Systems" *Proceedings of the IVHS AMERICA* conference.

Sedgewick, R. and J. S. Vitter, (1986), "Shortest Path In Euclidean Graphs" *Algorithmic* 1, pp. 31~48.

Tarjan, E. R., *Data Structure and Network Algorithms.* SIAM, Philadelphia, 1983.

Vuren, V. T. and G. R. M. Jansen, (1988), "Resent Developments In Path Finding Algorithms: A Review," *Transportation Planning and Technology.* Vol. **12**, pp 57~71.

CHAPTER 7

ESTIMATION OF DYNAMIC AND STOCHASTIC O-D TRAVEL TIME USING ARTIFICIAL NEURAL NETWORKS

7.0 INTRODUCTION

The travel time from one location (origin) to another (destination) in a traffic network, or O-D travel time, can be calculated exactly using shortest path algorithms as discussed in previous chapters. However, there are many situations which require a quick and accurate estimation of this O-D travel time. In these situations, even the heuristic algorithms presented in Chapter 6 may be not fast enough. For example, in a real-time vehicle dispatching system, a fleet of vehicles is required to be optimally routed and scheduled to visit a number of locations dispersed in the service area. During this routing and scheduling optimization procedure, the O-D travel times have to be calculated in thousands of times if the O-D travel time is modeled as a function of the departure time at the origin location. As a result, a shortest path algorithm, although most accurate, is not fast enough to be used in such types of applications. This fact is further demonstrated in Chapter 8.

The quick and accurate O-D travel time estimation may also be used to improve some of the heuristic shortest path algorithms discussed in Chapter 6. For example, it has been shown that a bi-directional shortest path algorithm is more efficient than a uni-directional algorithm, but the former algorithm is only applicable in networks where the travel times are static (Kuznetsov, 1993, Fu and Rilett, 1994). This is because in a dynamic network, the bi-directional algorithm requires exact information about the departure time at an origin node and the arrival time at a destination node. An accurate estimate of the minimum travel time may make it possible to implement a bi-directional algorithm in a dynamic network (Fu and Rilett, 1994).

O-D travel time can also be estimated based on the travel distance from the origin location to the destination location, or O-D travel distance and average travel speed. In this paper, this method is referred as to distance-based method. The O-D travel distance is commonly modeled as a function of location coordinates, or more often, a function of a family of distance functions (Love 1988). In most situations a linear regression model of rectangular distance and Euclidean distance is used to approximate the O-D travel distance. The estimation of the dynamic and stochastic O-D travel time in an urban environment, however, is a more complex problem than the estimation of the travel distance. This is because the O-D travel time is a function of road network topology, time of day, recurring/non-recurring congestion as well as travel distance. These factors all have a highly non-linear impact on the O-D travel time.

Artificial Neural Networks (ANN) have become one of the most popular techniques in the Artificial Intelligence (AI) field during the last decade. The special architecture and computation mechanics inherent in the ANN model make it useful for a wide variety of tasks such as image processing, pattern recognition and solving combinatorial problems. ANN have been found to be very useful in simulating the relationship between quantitative and qualitative inputs and their related output. A simplified description about ANN is provided in Appendix C and more detailed information may be found in other references (Rumelhart 1986).

The primary objective of this chapter is to demonstrate the feasibility of using an ANN for estimating dynamic and stochastic O-D travel times. The dynamic and stochastic attributes of the O-D travel time is assumed to be represented by the O-D travel time mean and standard deviation as a function of the time of day. Therefore, this research essentially tries to estimate the mean and standard deviation of the O-D travel time in a dynamic and stochastic urban traffic network. This research concentrates on modeling the O-D travel time associated with recurring congestion on a traffic network and therefore the estimated O-D travel time reflects historic traffic behavior. The applicability of ANN in the dial-a-ride vehicle routing and scheduling process is demonstrated in Chapter 8.

This chapter first proposes three feed forward neural networks to model the link travel time behavior (mean and standard deviation) during different time periods of a day: AM peak, PM peak and off peak. Subsequently, the chapter examines how the input attributes are identified, how the training data are represented and how the "best" ANN model is developed. The data for training and testing is simulated using a traffic network from the City of Edmonton described in Section 4.4.2 of Chapter 4 as a base. A comparison between the ANN models and the distance-based method is then presented. Lastly, the computation time of the proposed ANN model is compared to that of the exact shortest path algorithms.

7.1 NEURAL NETWORK BASED TRAVEL TIME ESTIMATION MODEL

7.1.1 ANN Network Topology

The ANN used in this analysis is called a back-propagation neural network and its topology is shown in Figure 7-1. The ANN consists of three layers with the neighboring layers fully connected. The output layer includes cells representing the variables to be estimated, that is, the O-D travel time. The input layer represents the factors which may have an impact on the O-D travel time, such as the origin and destination locations, the departure time at the origin and some other information. The next section provides a more detailed discussion on the selection of these factors. The number of hidden nodes is a decision variable and determined during the training and testing stage discussed in the following section.

It is well known that the travel times in a traffic network tend to be relatively stable, although the variability associated with these travel times can be high. In a typical urban environment there are three different time periods: the AM peak, the PM peak and the off peak, during which the travel time patterns differ significantly. In order to avoid the unnecessary training burden of using a single ANN to map the whole day travel time pattern, three separate ANN models are developed for these three time periods and are referred to as the AM Net, PM Net and OFF Net in this thesis. For the purposes of this research the AM peak is defined as lasting from 6 AM to 9 AM and the PM peak is defined as lasting from 3 PM to 7 PM. All other time periods belong to the off-peak period.

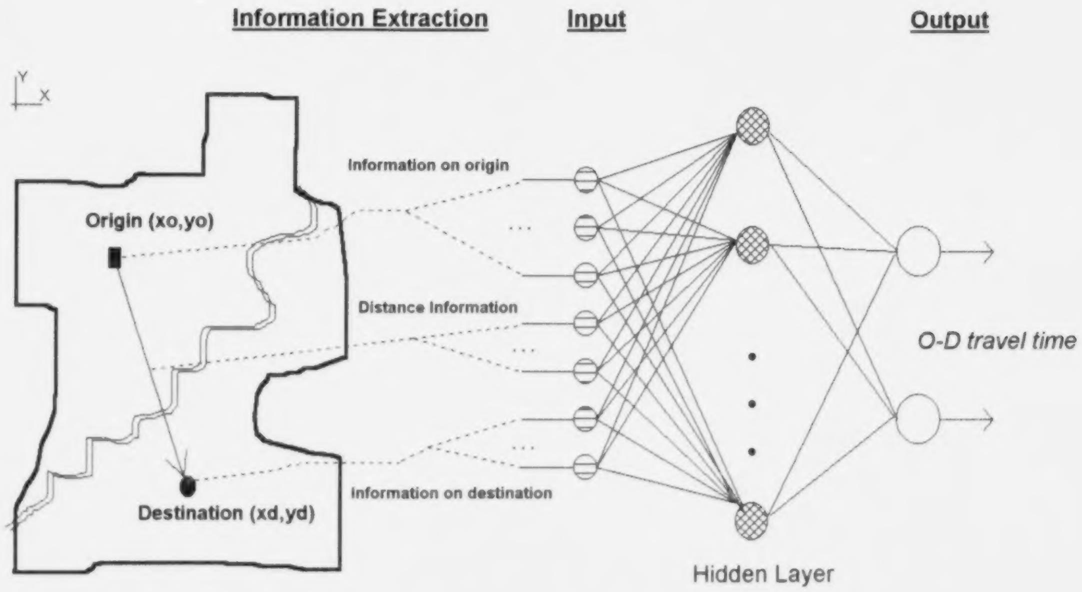

Information Extraction
Input
Output
Y
X
Origin (xo,yo)
Destination (xd,yd)
Information on origin
Distance Information
Information on destination
Hidden Layer
O-D travel time

Figure 7-1 ANN topology for O-D travel time estimation

7.1.2 Data Representation

The representation of the data is one of the most important steps in the development of a neural network model. There are two major steps in the data representation process. The first step identified the input and output attributes, which is schematically illustrated in Figure 7-1. The output, O-D travel time (t_{od}), is composed of the expected O-D travel time ($E[t_{od}]$) and the standard deviation of the O-D travel time ($S[t_{od}]$). There are two network design options with respect to output estimation. The first option (Option 1) is to use two separate neural networks, one for the mean estimation and another one for the standard deviation estimation. In this case, each network has one output cell. Another option (Option 2) is to estimate both output variables using a single network and consequently two output cells are needed. This research investigates both of these two options.

There are a wide variety of inputs and input combinations. In this research, two input scenarios are tested. Considering that the O-D travel time completely depends on the locations of the origin and destinations and the departure time, the first scenario considered (Scenario A) includes five input attributes: the coordinates of the origin and destination locations, i.e. (x_o,y_o) and (x_d,y_d), and the departure time(T_0), at the origin location.

Because the O-D travel time are directly related to the distance between the origin location and the destination location, two extra variables representing the estimated distance are added to those of scenario A to form the second scenario, i.e., Scenario B. These two variables are the rectangular distance (Manhattan distance) (ℓ_1) and the

Euclidean distance (ℓ_2) from the origin location to the destination location and are defined in (7-1) below.

$$\begin{aligned} \ell_1 &= |x_o - x_d| + |y_o - y_d| \\ \ell_2 &= \sqrt{(x_o - x_d)^2 + (y_o - y_d)^2} \end{aligned} \quad (7\text{-}1)$$

The second step involved data normalization. For the sake of learning effectiveness, all the input data are scaled into values that ranged between 0 and 1, while the outputs are scaled into values that ranged from 0.2 to 0.8. This ensures that the output remains in the quasilinear part of the sigmoid function where learning is faster (Gallant, 1993).

7.1.3 Training and Testing Examples

The training and testing data should be taken from the urban area where the O-D travel time estimation method is to be applied. It would be desirable if direct field collected O-D travel times are available for training and testing the ANN. Because there are no detailed data available for the modeling purposes of this research, simulated data are used for the purpose of this thesis. The simulated data are based on the Edmonton network described in Section 4.4.2 of Chapter 4. Consequently, the models developed in this study and their associated results are only valid for demonstration purposes, although the principles hold for more detailed networks used in conjunction with field data.

For each training and testing sample, the mean and standard deviation of the O-D travel time are calculated by applying a shortest path algorithm in the test network. These calculated O-D travel times (simulated) are assumed to be the real O-D travel time and therefore are referred as to *actual* O-D travel time in the following discussion. The dynamic and stochastic link travel time data for the Edmonton network are created as follows. The first step is to create the mean travel time profile for each link in the network. In order to consider the difference in the traffic patterns between different areas in the network, the urban area is divided into three sub-areas: Downtown, Mid-town and Suburban, as shown in Figure 7-2. The links in each area are assumed to have the same dynamic travel time profile (i.e., profile of mean link travel time) and these are shown in Figure 7-3 for each area. During peak hours the average travel time increases because of the increased volume in the network. The mean link travel time during the off-peak period is assumed to be equal to the link length divided by the posted speed. To model the inherent variability of the link travel time, the simulated mean link travel time is modeled as a uniformly distributed random variable. The formula for calculating this simulated travel time is shown in equation (7-2).

$$U_e = (0.95 + 0.10 \times \mathrm{U}\{0,1\})\, B_e \tag{7-2}$$

where U{0,1} = a uniformly distributed random variable with a range between 0 and 1.

U_e = mean link travel time for link e;

B_e = base link travel time for link e.

The second step is to simulate the variance of the link travel time. It is assumed that the coefficient of variation of the link travel times (COV), defined as the ratio of the standard deviation to the mean) are uniformly distributed and their distribution are only different by sub areas. Table 7-1 gives the hypothetical distribution of the link travel time COV for each subdivision. A larger value of COV is used for the links in the central area of the city considering that it has more traffic controls and a higher network density.

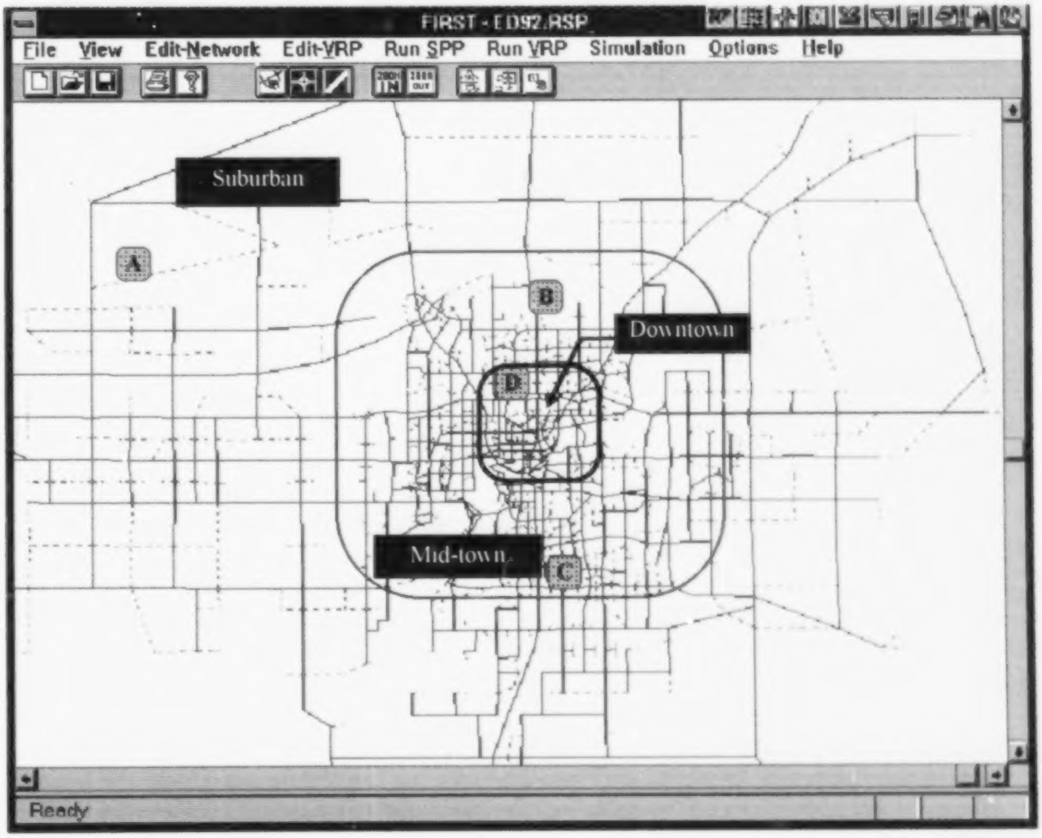

FIRST - ED92.RSP
File
View
Edit-Network
Edit-VRP
Run SPP
Run VRP
Simulation
Options
Help
Suburban
A
B
Downtown
D
Mid-town
C
Ready

Figure 7-2 The Edmonton network and sub-areas

Table 7-1 Link travel time covariance in different subareas

Subdivision	Downtown	Midtown	Suburban
Link Travel Time (COV)	U{0.9,1.0}	U{0.5,0.6}	U{0.1,0.2}

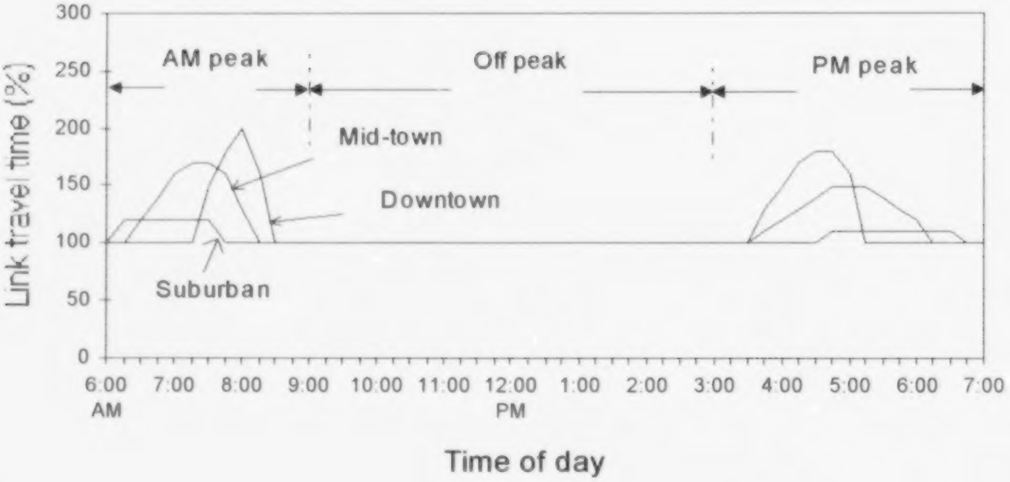


Figure 7-3 Dynamic link travel time pattern

The origin and destination locations are randomly generated and used to create the training data. For each O-D pair, six random departure times are generated during each of the AM, PM and off-peak periods, respectively. Given the departure time, the shortest path and the mean and standard deviation of the O-D travel time is calculated using the algorithms developed in Chapter 5. These data are then combined with the coordinate information for each O-D pair to form the training data. Table 7-2 shows an example of the training data for two O-D pairs.

Table 7-2 Training data examples (original data)

(X_o, Y_o) (m)	(X_d, Y_d) (m)	l_1 (m)	l_2 (m)	T_o* (Min)	$E[t_{od}]$ (Sec)	$S[t_{od}]$ (Sec)
(33814.1,34357.9)	(3011.3, 43866.8)	40312.	30796.5	394	3878	1820
(33268.2,37791.5)	(2891.0,37372.2)	32237.	30380.1	511	1419	786

Total minutes elapsed since midnight.

7.1.4 Training, Testing and Results

The training of the neural network is done via the back-propagation learning algorithm which is presented in Appendix C. The objective of the training and testing procedure is to find the best ANN topology to model the O-D travel time. The quality of an ANN is evaluated by two criteria. The first one is the learning speed which is reflected by the iterations needed to completely train the neural network. The second one is the mean square error (MSE) defined in the following equation (Gallant, 1993):

$$MSE = \sqrt{\frac{\sum_{k=1}^{N}\sum_{i=1}^{M}(Y_{ki} - D_{ki})^2}{N}} \qquad (7\text{-}2)$$

where N = total number of examples to be trained;

M = number of output cells;

Y_{ki} = actual value at output cell i for example k;

D_{ki} = estimated value at output cell i for example k.

Through the process discussed in Section 7.1.3, A total of 1000 data sets are generated as training examples and 250 data sets for testing. The procedure to identify the best ANN included three steps. The first step is to decide which input data scenario (Scenario A or Scenario B) gave the best results. For the specified purpose, this step considered the modeling quality of estimating the mean O-D travel time. It is found that the ANN with scenario B (7 inputs) is better than scenario A (5 inputs) in terms of both learning speed and prediction quality.

After selecting the input attributes, the second step identified the best representation of the location information. It is determined that the two distance measures contain most of the distance information between the two locations and that the coordinates only reflect the location information of the trip origin and destination. Therefore, the network area under analysis is divided into a 1000×1000 grid, and the O-D locations used coordinates based on this system. As expected, the use of exact coordinates is found to slow down the learning speed with only a very minor improvement in model quality.

The final step is to decide which network model of Option 1 (separated neural network model) and Option 2 (joined neural network model) discussed in Section 7.1.2 is more capable and to identify the optimum number of hidden nodes for each network model. In this step, neural networks with 2 to 20 hidden nodes are first trained and analyzed for both network models for the AM period. As expected, the joined neural network model required a more complex neural network (more hidden nodes) compared to the separate neural network model. It is found that for the separate neural network model, using 5 hidden nodes achieved better results than a neural network with 10 hidden nodes for the joined neural network model. Figure 7-4 and Figure 7-5 shows the estimated coefficient of variation (COV) of the O-D travel time and the actual O-D travel time COV as a function of trip length from both network models. The data are deliberately generated in that all the trips originate from a single location in the downtown. For the joined neural network model (Figure 7-4), it can be seen that the neural network over-estimates the O-D travel time variance for short trips, but provides good estimates for trips longer than approximately 10 minutes.

As compared to the joined neural network model, the separate neural network model provided much better results as shown in Figure 7-5. In both cases the O-D travel time COV decreases and approaches to stable values as the trip length increases. This result is expected because the link travel times COV are modeled high in central area and lower in surrounding areas for this study. It is therefore decided to use the separated neural network model in this thesis.

Next, neural networks with 2 to 20 hidden nodes are trained and analyzed for all three time periods. The neural network with 5 hidden nodes is found to be adequate to model both the AM and PM peak periods while the neural network with 4 hidden nodes is satisfactory for the off-peak period.

As an example, the MSE as a function of the number of iterations for the separated mean travel time AM Net (the so-called learning curve) is shown in Figure 7-6.

The neural networks are subsequently tested on 250 randomly generated O-D data sets. Figure 7-7 illustrates the results from a separate model with five hidden nodes for estimating the mean O-D travel time as compared to the actual mean O-D travel time during the AM peak period. The average relative error (the difference between the estimated mean travel time with the actual mean travel time divided by the actual mean travel time) is 12.1%. This relative error translates to about 264 seconds for the average trip length of 2180 seconds.

In order to show the ability of the trained neural networks to model the dynamics of the travel time during a day, the predicted mean travel and the actual mean travel time with different departure times of day are compared. Figure 7-8 shows these travel times for two O-D trips, where the estimated value is from a separate model. One trip is from the northwest of the Suburban (A) to the south of the Mid-town area (B) and another is from the south of the Mid-town area (C) to the Downtown (D) as shown in Figure 7-2. It can be found that the non-linear relationship between the O-D travel time with time of day is tracked relatively well by the neural networks.

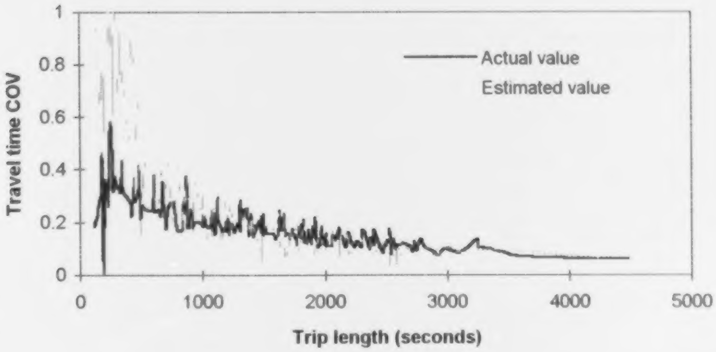


Figure 7-4 COV of the O-D travel time by a jointed network model: actual value vs. estimated value as a function of trip length

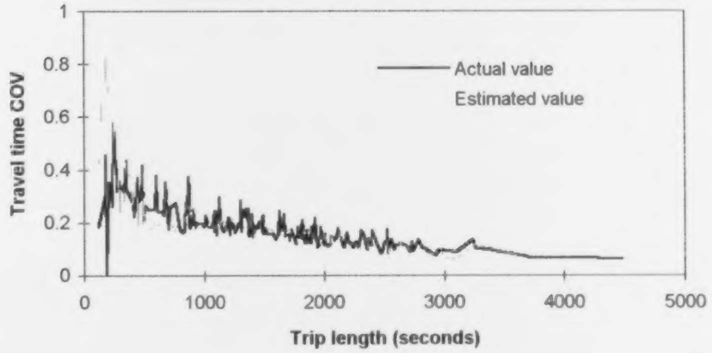


Figure 7-5 COV of the O-D travel time by a separate network model: actual value vs. estimated value as a function of trip length

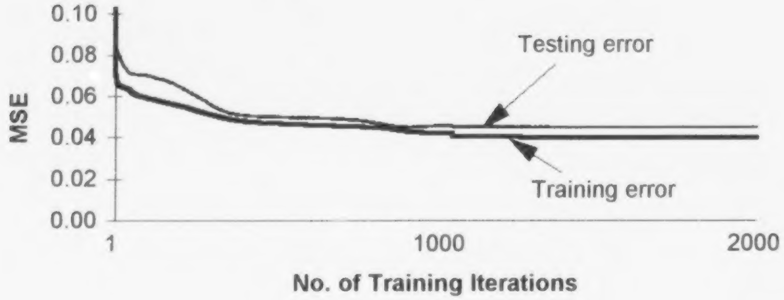


Figure 7-6 Learning progress curve for an AM net

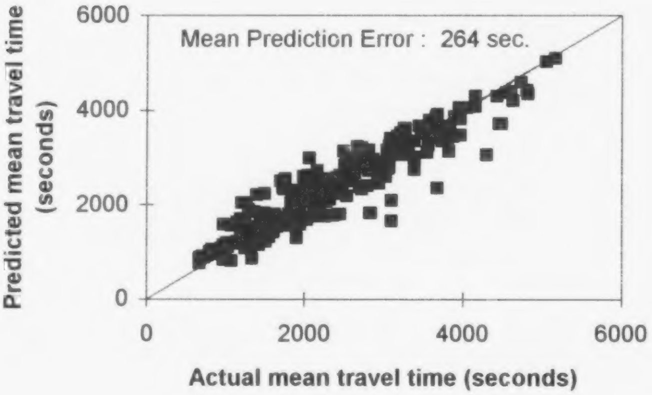
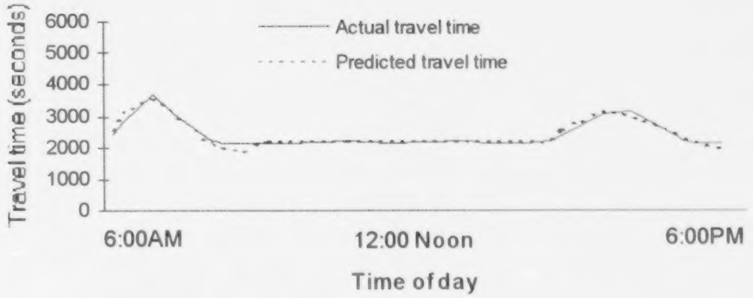


Figure 7-7 Actual mean travel time vs. travel time predicted by the AM net



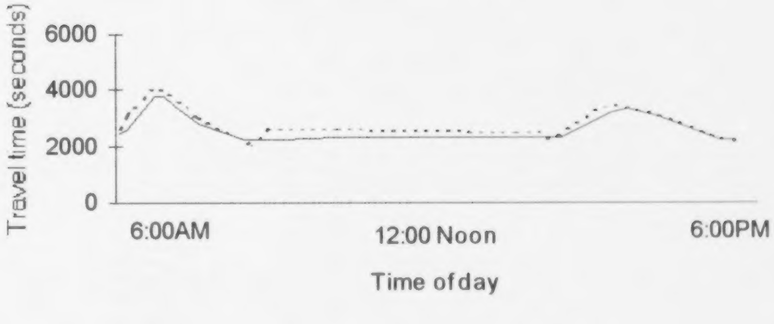


Figure 7-8 Actual travel time pattern vs. estimated travel time pattern: two O-D pairs

7.2 COMPARISON OF TRAVEL TIME ESTIMATION USING AN ANN AND A DISTANCE-BASED METHOD

The above sections showed how a ANN can be used to estimate the O-D travel time from one location to another in a traffic network. As discussed in Section 7.0, the relationship between the O-D travel time and location information can also be established using regression analysis. The objective of this section is to compare the relative performance between an ANN model and a distance-based model.

The off-peak period is deliberately selected as the modeling period so that the non-linear impact of the departure time on the O-D travel time can be removed. In addition, the travel times are assumed to be deterministic and therefore there is no O-D travel time variance needed to be modeled. As a result, the O-D travel estimation problem is then effectively the same as the O-D travel distance estimation problem in which distance-based models have been successfully applied(Love and *et. al.* 1988).

7.2.1 Data

A total of 1000 O-D sets are randomly generated for developing the regression model and new ANN models. In addition, 800 randomly generated O-D data sets are used for testing.

7.2.2 Models

The ANN used to model the off-peak period is the same as that shown in Figure 7-1 except that the departure time input cell and standard deviation output cell are removed.

The distance-based model included two variables: the rectangular distance (ℓ_1) and the Euclidean distance (ℓ_2), and is shown in the following equation:

$$t_{od} = 0.0262\,\ell_1 + 0.0237\,\ell_2 + 380.7, \qquad R^2 = 0.88$$

$$(9.151) \qquad (6.772) \qquad (23.966)$$

where:

t_{od} = travel time from origin(o) to destination(d), seconds;

ℓ_1, ℓ_2 = the rectangular distance and the Euclidean distance, refer to Equation (7-1) in Section 7.1.2

7.2.3 Results

Both the ANN and the distance-based model are applied to the test data and the results are summarized in Table 7-3. It can be seen that the relative estimation error from the neural network model is approximately 10 percent less than the regression model. It is anticipated that the dominance of the ANN over the distance-based model would be more significant if dynamic and stochastic travel times are modeled, that is, if the AM and PM peak times are used.

7.3 COMPARISON OF COMPUTATIONAL EFFICIENCY BETWEEN THE ANN METHOD AND SHORTEST PATH ALGORITHMS

The purpose of this section is to demonstrate the computational efficiency of the ANN models compared to the method of directly using the shortest path algorithm to calculate the travel time.

Table 7-3 Comparison of prediction error of ANN model and distance-based model

Data	Measurement	ANN Model	Regression Model
Modeling Data	Mean of the relative error	13.7%	20.7%
	Standard deviation of the relative error	14.0%	31.8%
Testing Data	Mean of the relative error	18.1%	29.4%
	Standard deviation of the relative error	25.8%	31.3%
Notation:	Relative error = |predicted value-actual value|/actual value		

The comparison is based on the Edmonton road network as shown in Figure 7-2. The label setting algorithm (LS) and the A* algorithm presented in Chapter 6 are used to find the expected minimum O-D travel times. One thousand O-D pairs are randomly generated and their travel time are found by using LS, A* and a trained ANN. In addition, the CPU times for each algorithm are recorded. It should be noted that all the programs in this study are coded in C++ and on a 486 computer with a 50 MHz clock speed and 8 MB RAM. It can be seen that the ANN is approximately 800 times faster than the LS algorithm and 500 times faster than the A* algorithms.

Another advantage to the ANN is that the computation time of an ANN does not change as O-D travel time increases. In contrast to the ANN model, the computation effort of a shortest path algorithm increases exponentially as the O-D travel time increases. Figure 7-9 shows the computation time of the A* as a function of O-D travel time for the Edmonton network.

Table 7-4 Computation time of the shortest path algorithms and the ANN model

Method	Total CPU to calculate the travel times of 1000 O-D pairs (Seconds)
LS algorithm	390
A* algorithm	230
ANN model	0.49

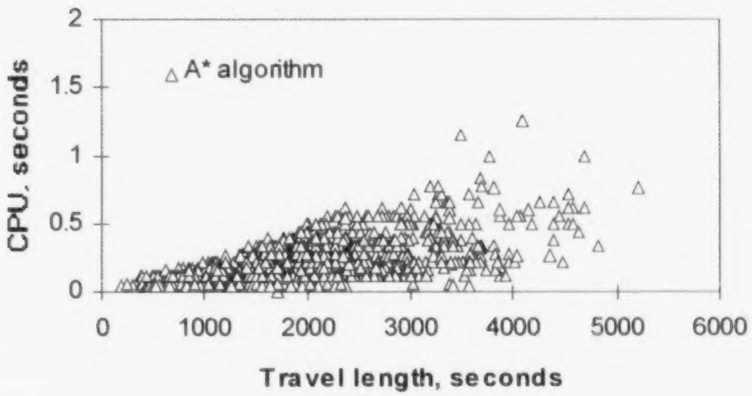


Figure 7-9 The relationship between the computation time and the travel time: A* shortest path algorithm

7.4 CONCLUSIONS

This chapter introduced the concept of using ANN models for estimating the dynamic and stochastic O-D travel time in a urban traffic network. Based on real network data and simulated link travel time pattern, variety of ANN models are trained and tested. The following conclusions may be drawn:

1. It is found that an ANN model can be trained to effectively map the highly non-linear relationship between the O-D travel time and their location information in dynamic and stochastic traffic networks;
2. The success of an ANN technique for travel time estimation mainly depends on how the input information is abstracted and what type of network model is used. This study demonstrated that some enhanced data (for example, distance information) can be very helpful in improving the performance of an ANN, and separate models for different parameters to be estimated are much more effective than using a joined network model.
3. The solution quality of the ANN method is found to be significantly better than the traditional regression model for estimating O-D travel times. It can therefore be expected that great savings can be obtained in the applications by using the ANN O-D travel time estimation method instead of the traditional method;
4. While the ANN is not as accurate as the shortest path algorithms, it is much faster. It has been empirically shown that the ANN is more than 500 times faster than the shortest path algorithms. Therefore it is useful in situations where travel time

calculations are necessary but where the computation time is limited. It can be expected that the ANN model holds great potential to be applied in various real-time on-line applications such as AVDS. Chapter 8 further discusses how to integrate the ANN models in the dial-ride vehicle routing and scheduling process.

REFERENCES:

Kuznetsov, T., (1993), "High Performance Routing for IVHS," IVHS America 3rd Annual Meeting, Washington, D. C. April, 1993.

Fu L. and L. R. Rilett, (1994), "Neural Network Based Bi-directional Shortest Path Algorithm in A Dynamic Road Traffic Network," Working paper, University of Alberta, 1994.

Love, R. F., J. G. Morris and G. O. Wesolowsky, *Location: Models & Methods Facilities*, North-Holland, 1988.

Rumelhart, D. E. and McClelland, J. L. (eds.), *Parallel Distributed Processing: Explorations in the Micro structure of Cognition*, Vol. 1, MIT Press, Cambridge, 1986.

Gallant, S. I., *Neural Network Learning And Expert Systems*, The MIT Press, Cambridge, Massachusetts, London, England, 1993.

CHAPTER 8

DIAL-A-RIDE VEHICLE ROUTING AND SCHEDULING WITH DYNAMIC AND STOCHASTIC O-D TRAVEL TIME

8.0 OVERVIEW

As discussed in previous chapters, the travel time from one location (origin) to another location (destination), or O-D travel time, is dynamic and stochastic due to the inherent variation of urban traffic congestion, weather conditions, and even driving behavior. For simplicity, the O-D travel time has been assumed to be static and deterministic in most existing models of the dial-a-ride vehicle routing and scheduling problem (DARP) (Bodin *et al.*, 1983; Sexton and Bodin,1985a,1985b; Wilson *et al.*, 1977; Psaraftis, 1983; Jaw *et. al.* 1986; Savelsbergh and Sol, 1995). It can be expected that in situations of high uncertainty the service vehicles may not be able to follow the schedules found based on these models and thus a reliable service may not be guaranteed. For example, based on the assumption of deterministic O-D travel time it would be feasible to schedule a vehicle to drop off a customer at his/her destination at his most desired drop-off time. However, there is a certain amount of uncertainty that the customer may be dropped off after his most desired time because of the randomness of the vehicle travel times. The drawback associated with the assumption of static O-D travel time is more straightforward in the sense that the use of an incorrect O-D time would result in erroneous and inefficient schedules. The first objective of this chapter is to develop a dial-a-ride vehicle routing and scheduling model which explicitly considers the dynamic and stochastic attributes of the O-D travel time and to analyze the impact of the dynamic and stochastic attribute of the O-D travel time on the routing and scheduling results of dial-a-ride vehicles.

As previously discussed in Chapter 7, the O-D travel time can be estimated using various methods such as the distance based method and the ANN method as presented in Chapter 7, and the shortest path algorithms as discussed in Chapter 6. These methods have been shown to be very different in terms of estimation quality (or accuracy) and computational efficiency which in turn influences the performance of the routing and scheduling algorithm. The second objective of this chapter is to investigate the feasibility of using these O-D travel time estimation methods in the dial-a-ride vehicle routing and scheduling algorithm.

This chapter first discusses how the DARP can be modeled with respect to the objectives of the system operator and customers when the O-D travel times are modeled as random variables and a mathematical formulation of the new DARP is subsequently provided. Next, a heuristic dial-a-ride vehicle routing and scheduling algorithm is introduced to solve this problem. Lastly, a computational study is then conducted to illustrate the difference in solutions between the models with and without considering the randomness of the O-D travel time and the computational efficiency of the proposed algorithm when different O-D travel time estimation methods are used.

8.1 DIAL-A-RIDE VEHICLE ROUTING AND SCHEDULING PROBLEM: MODELS

There are fundamentally two vehicle routing and scheduling problems involved in a dial-a-ride service system that need to be treated differently. The first problem is the *static* DARP, or *subscriber* DARP as referred to in this thesis, which usually needs to be solved at the beginning of every day when all the customer requests are known (for example, booked one day in advance, these trips are also called subscribed trips). The objective of the routing and scheduling procedure is to determine the assignment of all customers (or trips) to the available vehicles and their respective routes and schedules.

The second problem in a dial-a-ride system is called *dynamic* DARP, or *real-time* DARP as referred to in this thesis, in which the objective is to determine the assignment of a new customer into the existing schedule of a vehicle in real-time (these trips are also called demand trips). The new customer usually phones the service center to request an immediate service while each fleet vehicle is already in service and has been given an operation schedule. The problem is computationally tractable when only one vehicle's schedule is allowed to be changed and the original visiting order of the vehicle to be changed must be maintained. The methods to solve this problem can be readily extended from the algorithms for the advance request dial-a-ride problem (Bodin *et al.*, 1983). However, for real-time operation purpose, the dispatcher has to immediately give the customer his/her pickup/dropoff time; therefore, the "optimal" insertion schedule must be found in a very short time period (for example, a few seconds). This real-time operation requirement may hinder the use of more realistic, but more time consuming models and algorithms. While the feasibility of solving this real-time problem with different O-D travel time estimation methods is illustrated in Section 8.3.4, the modeling methodology and algorithm are described through the *subscriber* DARP.

In order to model and formulate the DARP it is necessary to take into account both the service operator's objective and the customers' objectives. The term objective, as discussed in this thesis, means both variable objectives (for example, minimize total travel time) and fixed objectives or constraints (for example, customer's drop-off time must be earlier than his/her desired drop-off time). Due to the involvement of the multiple variable objectives and the randomness of the system status (i.e. random O-D travel time), the utility concept is introduced to combine the variable objectives of the system operator and the customers, and to resolve the randomness of the operation measure (Keeney and Raiffa, 1976). Specifically, a disutility function, which represents the relationship between the degree of dissatisfaction of the service operator or customers associated with the routing and scheduling results, is used to represent the variable objectives. Therefore, the general objective of the routing and scheduling procedure can be represented by the total disutilities of the service operator and customers. In the following sections, the objectives (i.e., disutility functions and constraints) related to the service operator, drivers and customers are defined when the O-D travel times are random. It should be noted that because it is not the goal of this thesis to examine what is the actual format of the utility function for modeling the service operator or customers' risk attitude, some commonly used utility functions are adapted in the following discussions.

8.1.1 Objectives Related to Service Operator

The service operator's objective is to provide transportation service to customers while minimizing the total operating cost. The operating cost is commonly assumed to be proportional to the total travel time and the number of vehicles used (Savelsbergh and Sol, 1995). This latter consideration is especially important when some rented or contracted vehicles (in addition of internal vehicle fleet) have to be used in service. This thesis assumes that there is a certain number of vehicles available for service. Therefore, the objective is to minimize the total travel time given a fixed number of service vehicles.

When the travel time is random, the objective of minimizing the total travel time needs to be redefined. By assuming that the operator is risk neutral in terms of travel time, the expectation of the total travel time can therefore be used to represent the operator's disutility, DU_o:

$$DU_o = a_1 E[tt] \tag{8-1}$$

where: tt = a random variable representing the total vehicle travel time;

$E[tt]$ = expectation of the total vehicle travel time tt;

a_1 = a constant representing the weight allocated to the operator's objective in the general objective of the routing and scheduling procedure.

8.1.2 Objectives Related to Each Service Vehicle and Driver

The operational objectives related to service vehicles or drivers could include the preferred or pre-specified service area and working shift of each driver (length of working period and break time). These objectives are considered as constraints in the routing and scheduling process. This thesis only considers vehicles' working time period.

8.1.3 Objectives Related to Customers

The customers' objectives are reflected in the customers' satisfaction from the provided service, which can be reflected by two measures. One measure is the time deviation from the customer's desired pick-up/ drop-off time which represents the closeness between the actual or scheduled pick-up/drop-off time and his/her most desired pick-up/drop-off time. The other measure is the customer's excess ride time as compared to his/her direct ride time (without diversion). In the routing and scheduling process, both of these two measures need to be explicitly considered. The following section provides some detailed discussions on how these objectives are modeled when the travel time and arrival time are random.

8.1.3.1 Satisfaction from the pick-up/drop-off time

It is assumed that each customer has a desired pick-up/ drop-off time. This desired pick-up/ drop-off time indicates that he/she should not be scheduled to be picked-up (dropped-off) earlier (later) than this time. In addition, the customer may also expect that the actual service time is as close to his/her desired time as possible. In order to avoid an excessively large deviation from the desired time, a maximum allowable deviation value is usually set in the scheduling process. Therefore, a closed time window can be formed for each customer to specify the time period in which a customer must be picked up (dropped off). If a customer specifies a desired pick-up time, there is a pick-up time window for him/her. Otherwise, there is a drop-off time window for him/her.

In this thesis, the travel time is assumed to be random, therefore, the vehicle arrival time at a stop (i.e., pick-up/ drop-off time) must also be modeled as a random variable. This implies that each customer must be dropped off/picked up within their individual time window with a given probability. The time constraints used in deterministic models need to be modified to reflect the stochastic attributes of service time. For example, for a customer who specifies a desired drop-off time, the modified constraints should stated that the customer must be dropped off at his/her destination within his/her desired drop-off time period (time window) within a given probability. This thesis uses Equation (8-2) to represent this type of new constraint for each pick-up/drop-off operation.

$$\Pr ob(ET_i \leq T_i \leq LT_i) \geq \alpha \qquad (8\text{-}2)$$

Where: ET_i = the earliest time to pick up/ drop off customer i;

LT_i = the latest time to pick up/drop off the customer i;

α = a pre-specified constant representing the minimal probability.

Since each customer is allowed to specify either a desired pick-up time or desired drop-off time, the above constraint (either pick-up time or drop-off time) needs only be tested once for each customer. For a customer who specifies a desired pick-up time (τ_i), the pick-up location has a time window with $ET_i = \tau_i$ and $LT_i = \tau_i + \delta$, where δ is the maximum deviation from the desired time allowed. For a customer who has a desired drop off time (τ_i), only the drop-off time needs to be tested with $ET_i = \tau_i - \delta$ and $LT_i = \tau_i$. The calculation of the probability from Equation (8-2) requires information on the distribution of the vehicle arrival time. This thesis assumes that the vehicle arrival time at a stop is a continuous random variable and its PDF is noted as $f_{T_i}(x)$. Therefore, Equation (8-2) can be rewritten as shown in Equation (8-3).

$$\int_{ET_i}^{LT_i} f_{T_i}(x)\,dx \geq \alpha \tag{8-3}$$

Figure 8-1 schematically illustrates this type of constraint where the curve represents the vehicle's arrival time (pick-up time/ drop-off time) PDF, and the shaded area under the curve is the probability that the customer may be picked up or dropped off within the time window $[ET_i, LT_i]$.

It should be noted that two issues need to be addressed before Equation (8-3) may be used in a scheduling algorithm. The first issue is how to determine the PDF of the vehicle arrival time at a stop (pick-up or drop-off location). The second issue is how to calculate the integration in Equation (8-3) when the PDF is given. Similar to the argument presented in Chapter 4, it is unfeasible to determine the PDF of the vehicle arrival time at each stop when the O-D travel times are dynamic and stochastic. Furthermore, even if the PDF can be identified, the integration involved would impose a large computational burden for a routing and scheduling algorithm. In order to avoid these problems, the vehicle arrival time at each stop is assumed to be normally distributed. Consequently, the problem of estimating the PDF becomes a problem of estimating its mean and variance which can be solved using the approximation models developed in Chapter 4 and further explored in Chapter 5 and Chapter 7.

The integration of a normally distributed density function can be accurately approximated by some explicit expressions or an explicit lookup table, and thus the computational burden may be greatly reduced. This thesis used the formula shown in Equation (8-4) to approximate the distribution function of a standard normal distribution (Karian and Dudewicz, 1991).

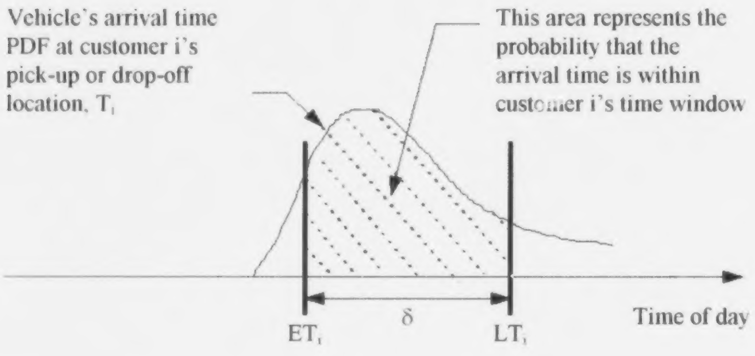

Vehicle's arrival time PDF at customer i's pick-up or drop-off location, T_i
This area represents the probability that the arrival time is within customer i's time window
δ
ET_i
LT_i
Time of day

Figure 8-1 Time window with random arrival time

$$\Phi(x) = \int_{-\infty}^{x} f_X(x)\,dx = \frac{x(4.4 - x)}{10} + 0.5 \qquad (8\text{-}4)$$

Although the time window represented in Equation (8-2) provides a general control on the service quality, it does not completely reflect customers' satisfaction or service requirements. For example, a customer who specified a desired drop-off time not only wants to be dropped off at his/her destination before his/her most desired time, but also expects that his/her drop-off time is as closer to his/her most desired time as possible. The larger the time deviation, the less satisfied the customer would be. In order to include this type of customer dissatisfaction in the scheduling process, a disutility is introduced as part of the objective function. For customer i, the disutility due to the deviation from the most desired pick-up and drop-off time, Du_i^d, is defined as follows:

$$DU_i^d = \int_{-\infty}^{\infty} f_{T_i}(\tau_i + x)\cdot U(x)\,dx \qquad (8\text{-}5)$$

Where:

x = deviation from the desired time defined as:

$$x = \begin{cases} T_i - \tau_i & \text{For customers specifing a desired pick-up time} \\ \tau_i - T_i & \text{For customers specifing a desired drop-off time} \end{cases}$$

$f_{T_i}(x)$ = the probability density function of the arrival time T_i;

$U(x)$ = a utility function representing a customer's dissatisfaction from the schedule as a function of the deviation from his/her desired time (x).

A general form of the utility function which is commonly used to model customers' risk-aversion attitude under uncertainty is the quadratic function (Wilson *et al.*, 1977; Jaw *et. al.* 1986). To simplify the integration involved in Equation (8-5), the linear term of the general quadratic function is taken out and the resulting utility function is $U(x) = a_2 x^2$, where a_2 is an externally specified constant indicating how much weight the importance of the time deviation is allocated in the general objective function.

Once the disutility function is identified, the disutility due to time deviation shown in Equation (8-5) can be transformed as a function of the mean and variance of the arrival time at the pickup/drop-off stop, as shown in Equation (8-6):

$$DU_i^d = a_2 \cdot (E[T_i] - \tau_i)^2 + a_2 \cdot Var[T_i] \qquad (8\text{-}6)$$

From the Equation (8-6), it can be seen that the disutility due to time deviation not only depends on the expected deviation from the most desired time ($E[T_i]-\tau_i$), but also on the variance of the arrival time ($Var[T_i]$). This relationship makes intuitive sense in that it would be expected that due to arrival time variation a customer may not be satisfied even when the expected arrival time deviation is zero.

8.1.3.2 Satisfaction from the ride time

In addition to the service time deviation, the other important measure of service quality is the customer's ride time. As a shared ride system, customers may experience excess ride time (difference between actual ride time and direct ride time) and the larger the actual ride time for a customer, the more dissatisfied the customer will be. In order to avoid too much excess ride time in a schedule, most dial-a-ride systems specify a maximum ride time as a criteria. When the travel time is random, this condition can be expressed as a constraint involving a probability function as shown in Equation (8-7):

$$\Pr ob(\tilde{t}_i \leq L) \geq \beta \tag{8-7}$$

Where $\tilde{t}_i$ is a random variable representing the scheduled ride time for customer i and β is an externally set criteria. For example, it may be decided that the constraint must be satisfied with a probability of 95%. This constraint is schematically illustrated in Figure 8-2, where the curve represents the customer's ride time PDF and the shaded area under the curve is the same probability as shown in Equation (8-7). Equation (8-7) can be rewritten as shown in Equation (8-8):

$$\int_0^L f_{\tilde{t}_i}(x)dx \geq \beta \tag{8-8}$$

Where: $f_{\tilde{t}_i}(x) =$ the PDF of customer i's ride time.

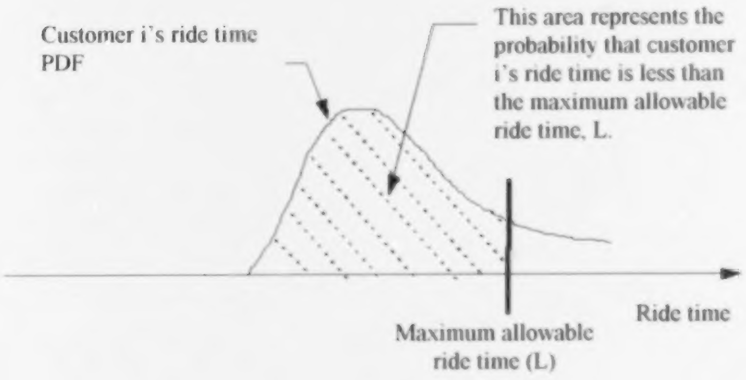

Customer i's ride time PDF
This area represents the probability that customer i's ride time is less than the maximum allowable ride time, L.
Maximum allowable ride time (L)
Ride time

Figure 8-2 Customer's ride time condition

Similar to the treatment for the time deviation condition discussed above, customer's ride time is assumed to be normally distributed and the integration involved in Equation (8-8) can be calculated based on Equation (8-4).

The maximum ride time criteria discussed above provides an upper bound of the service quality. However, it does not reflect the customer's desire of having their ride diversion as small as possible. Therefore, a new disutility item is incorporated in the general objective function. Considering that both actual ride time and direct ride time are random variables, the new disutility function uses the relative mean ride time diversion as an indicator of customers' dissatisfaction. For illustration purpose, this thesis uses a quadratic function as shown in Equation (8-9), to represent each customer's utility function.

$$DU_i^r = a_3 y_i^2 \tag{8-9}$$

Where: a_3 = an externally set constant representing the weight allocated to the ride diversion in the general objective function;

y_i = the relative deviation of mean ride time for customer i, defined as:

$$y_i = \frac{E[\tilde{t}_i] - E[\bar{t}_i]}{E[\bar{t}_i]} \tag{8-10}$$

Where: $\bar{t}_i$ = the direct ride time with expected direct ride time noted as $E[\bar{t}_i]$;

$E[\tilde{t}_i]$ = the expected scheduled ride time for customer i;

The reason for using the relative ride time diversion function shown in Equation (8-10) is that a customer's toleration on the ride time deviation is also related to the direction ride time of his/her trip. Commonly, the longer the direction ride time of his/her trip is, the larger ride time deviation he/she can tolerate.

8.1.4 Problem Formulation

The previous sections have discussed how the service operators' objective and customers' service needs can be modeled when the O-D travel times are dynamic and stochastic. To clarify the overall problem structure, the *subscriber* DARP with dynamic and stochastic O-D time is formulated as a mathematical program.

Let M be the set of vehicles. Each vehicle $k \in M$ has a seat capacity Q_k, a start location $s_k \in S$ and an end location $e_k \in E$. Each vehicle is assumed to start empty. Let N be the set of trip requests. For each trip request $i \in N$, a load of size (in terms of seats required) q_i has to be transported from an origin location $o_i \in O$ to a destination location $d_i \in D$. For each location $i \in O \cup D$ there is a desired time window with an earliest service time ET_i or/and latest service time LT_i.

For all $i,j \in S \cup E \cup O \cup D$ let $\{t_{ij\,Ti}, T_i\}$ denote the stochastic process of the travel time from location i to location j with departure time at node i equal to T_i. It should be noted that this definition is similar to the definition of the dynamic and stochastic link travel time provided in Section 4.1 of Chapter 4. For any given T_i, $t_{ij\,Ti}$ is assumed to be normally distributed with a known mean noted as $\mu_{(Ti)}$ and variance noted as $\sigma^2_{(Ti)}$ (It should be noted that when denoting the mean and variance, the subscript $_{ij}$ is deliberately omitted for notation convenience). Note that the loading time (pickup/drop-off time) at the origins and destinations can be easily incorporated in the travel time and is not considered explicitly. It is also assumed that vehicles are not allowed to idle at each stop or on road.

The following four types of decision variables are introduced to represent the vehicle schedules:

$$x_{ijk} = \begin{cases} 1 & \text{if vehicle k visits location j immediately after location i;} \\ 0 & \text{otherwise} \end{cases}$$

$$y_{ik} = \begin{cases} 1 & \text{if trip request i is assigned to vehicle k;} \\ 0 & \text{otherwise} \end{cases}$$

T_i = arrival time at location i, a random variable;

v_i = the load of the vehicle after visiting location i;

The DARP may be formulated as follows:

Objective: minimize a total cost function which is a summation of the system operator's disutility (DU_0) defined in Equation (8-1) and the customers' disutility defined in Equation (8-5) and (8-10) as follows:

$$\textit{Minimize} \qquad C = DU_o + \sum_{i}^{N} DU_i^d + \sum_{i}^{N} DU_i^r \tag{8-11}$$

Subject to the following constraints:

Constraint 1: each customer is only serviced by one vehicle:

$$\sum_{k \in M} y_{ik} = 1 \qquad (\text{ for all } i \in N\ ) \tag{8-12}$$

Constraint 2: vehicle load and capacity related:

$$v_i = 0 \qquad (\text{ for all } i \in S) \tag{8-13}$$

$$v_j = v_i + q_j \qquad (\text{ when } x_{ijk} = 1,\ \text{for all } i,j \in S \cup E \cup O \cup D,\ k \in M) \tag{8-14}$$

$$v_i \le \sum_{k \in M} Q_k y_{ik} \qquad (\text{ for all } i \in O \cup D) \tag{8-15}$$

Constraint 3: each stop (pick-up or drop-off location) is visited once:

$$\sum_{j \in S \cup E \cup O \cup D} x_{jik} = \sum_{j \in S \cup E \cup O \cup D} x_{ijk} = y_{ik} \qquad (\text{ for all } i \in O \cup D,\ k \in M\ ) \tag{8-16}$$

Constraint 4: relation between the departure time (mean and variance) of vehicles at locations can be obtained using the approximation models developed in Chapter 4 (refer to Equation (4-8) and Equation (4-12) in Chapter 4):

$$E[T_j] \cong E[T_i] + \mu(E[T_i]) + \mu''(E[T_i])\, Var[T_i]/2$$

$$Var[T_j] \cong \{1+\sigma'^2(E[T_i])+2\mu'(E[T_i])+\mu'^2(E[T_i])\}\, Var[T_i]+\sigma^2(E[T_i])$$

$$(\text{ when } x_{ijk} = 1, \text{ for all } i,j \in S \cup E \cup O \cup D,\ k \in M) \qquad (8\text{-}17)$$

Constraint 5: customers' desired service time windows (same as Equation (8-2) shown in section 8.2.4.1):

$$\Pr ob(ET_j \le T_j \le LT_j) \ge \alpha \qquad (\text{ for all } j \in O \cup D) \qquad (8\text{-}18)$$

Constraint 6: customers' maximum ride time (same as Equation (8-7) shown in Section 8.2.4.2):

$$\Pr ob(\tilde{t}_i \le L) \ge \beta \qquad (\text{ for all } i \in N) \qquad (8\text{-}19)$$

Constraint 7: general

$$x_{ijk} \in \{0,1\} \qquad (\text{ for all } i \in S \cup E \cup O \cup D,\ k \in M) \qquad (8\text{-}20)$$

$$y_{ik} \in \{0,1\} \qquad (\text{ for all } i \in N,\ k \in M) \qquad (8\text{-}21)$$

$$T_i \ge 0 \qquad (\text{ for all } i \in S \cup E \cup O \cup D) \qquad (8\text{-}22)$$

$$v_i \ge 0 \qquad (\text{ for all } i \in S \cup E \cup O \cup D) \qquad (8\text{-}23)$$

As compared to the deterministic DARP discussed in Chapter 2, the major extension of the dial-a-ride problem formulated above is modeling the O-D travel time as a stochastic process. The deterministic DARP is therefore a special case of this new formulated dial-a-ride problem. The former has been proved to be computationally intractable (or NP-complete), therefore, the stochastic dial-a-ride problem must also be

NP-complete and any optimal solution method to this problem may only be viable to small size problems (for example, less 50 customer trips). Since this thesis concerns itself mainly with the development of solution methods to the large size problems from practical applications (for example, over thousand customer trips), a heuristic algorithm is used to solve the problem formulated above.

8.2 HEURISTIC DIAL-A-RIDE VEHICLE ROUTING AND SCHEDULING ALGORITHM

Due to its inherent intractability, the subscriber DARP is often solved by heuristic algorithms for practical applications (Savelsbergh and Sol, 1995). There are typically two types of heuristic routing and scheduling procedures widely used: a sequential insertion procedure and a concurrent insertion procedure (Bodin *et al.*, 1983). These algorithms can be extended to solve the dial-a-ride problem with the new objective functions and constraints discussed above. For the objective of this thesis, the concurrent algorithm is extended in this chapter. The following is a generic version of this solution procedure:

Step 1: Determine the time window for the pick-up and drop-off of each customer;

Step 2: Select a customer i from the customer list.

Step 3: For each vehicle k from the fleet:

(1). find all feasible ways in which customer i can be inserted into the partial schedule of vehicle k. If it is unfeasible to insert customer i into vehicle k, examine the next vehicle k+1 and restart step 3;

(2) find the insertion of customer i into vehicle k which results in a minimum insertion cost;

Step 4: If it is not feasible to assign customer i to any vehicle, then either hire a new vehicle to serve this customer or "reject" this customer. Otherwise, assign customer i to vehicle k* for which the insertion cost is minimal among all the vehicles.

Central to the concurrent insertion algorithm are two steps: a *feasibility test* step to insert a customer trip into an existing schedule and an *optimization* step to find the best feasible insertion and best schedule. A detailed discussion of these two steps are provided in the following sections. In addition, how various O-D time estimation methods can be integrated into the above algorithm is discussed.

8.2.1 Feasibility Test

The feasibility test is to assure that the service quality constraints for both the newly inserted customer and all other customers already on that vehicle are not violated. The following tests must be passed for the feasibility check:

Test 1: test the vehicle's load (on board passengers) at each stop. The vehicle's load must always be less or equal to the vehicle's capacity as expressed in Equation (8-15);

Test 2**:** test the satisfaction of the inserted customer and the customers already on the vehicle. This test includes the customers' time window (Equation (8-18)) and the customers' ride time (Equation (8-19)).

8.2.2 Optimization

The optimization step is to minimize the total additional cost due to inserting a customer into a vehicle's schedule. The cost function is defined in Equation (8-11) and the additional cost is the difference between the cost before and after inserting a customer. As discussed in previous sections, the cost function is primarily related to three measures which include the total travel time for each vehicle, the arrival time at each stop and each customer's ride time. Once a vehicle's visiting sequence is determined (for example, after a customer is inserted into an existing partial route), an optimal starting time for this vehicle can be obtained by minimizing the total cost (or total disutility) of the schedule based on the O-D travel time information. Subsequently, the total travel time, the arrival time at each location and each customer's ride time can be determined to arrive at the total cost.

8.2.3 O-D Travel Time Estimation Methods

As seen in the previous section, the O-D travel time is the key information needed for the dial-a-ride vehicle routing and scheduling process. The estimation quality and computational efficiency of the O-D time estimation method directly influence the performance of the dial-a-ride vehicle routing and scheduling process. In order to analyze the feasibility of integrating the O-D travel time estimation methods, as developed in previous chapters, into the dial-a-ride vehicle routing and scheduling process, a computational analysis is conducted in Section 8.3.4 which examines the following three O-D travel time estimation methods: a) *Distance based method*: Discussed in Chapter 7, b) *ANN method*: Discussed in Chapter 7, c) *Heuristic shortest path algorithm:* Discussed in Chapter 6.

8.3 COMPUTATIONAL ANALYSIS

In the previous section a heuristic algorithm has been proposed to solve the DARP problem presented in Section 8.1. The objective of this section is to demonstrate the solution and computational requirements of the proposed algorithms as compared to the traditional deterministic model through simulated problems. The proposed algorithm is first applied to solve the simulated problems under various model parameter settings (Section 8.3.2 and Section 8.3.3) and using various O-D travel time estimation methods (Section 8.3.4). The results are then used to illustrate the influence of O-D travel time variations and O-D travel time estimation methods on the system performance. The system performance is measured using the following four statistics:

a) *Number of Vehicles Required*: Total number of vehicles required to serve the transportation requests;

b) *Expected Vehicle Productivity (trips/vehicle hour)*: defined as the ratio of total number of customers (or trips) to the total expected vehicle time;

c) *Expected Average Excess Ride Time (minutes)*: defined as the total expected excess ride time of all customers divided by total number of customers;

d) *Expected Average Time Deviation (minutes)*: defined as the total expected time deviation divided by total number of customers.

8.3.1 Test Problems

Two test problems are created for examining the performance of the model and algorithms presented in this chapter. The first one is used to investigate the difference in solutions obtained from solving the subscriber DARP with and without considering the stochastic attribute of the O-D travel time. The following is an assumed operation scenario for this problem:

a) The service area is a square of 20x20 square kilometers;

b) The O-D travel time is assumed to be normally distributed with a mean equal to the ratio of the rectangular distance to an average travel speed of 40 km/h. The coefficient of variation of the O-D travel time (equal to mean/standard deviation) is assumed to be same for all O-D pairs;

c) There are a total of 50 vehicles available and all of them are located in the center of the service area. All vehicles have the same seat capacity of 8 seats/vehicle. Each vehicle's available service time is from 6:00AM to 6:00PM;

d) There are a total of 200 customers (trips) uniformly located in the service area. Each customer either has a desired pick-up time or a desired drop-off time (50 percent each in this study) with a service time between 8:00AM and 12:00AM. The maximum time deviation from each customer's desired time is set to 30 minutes and the maximum ride time is set to 90 minutes.

The second problem is used to examine the computational efficiency of the proposed algorithm when using different O-D travel time estimation methods. This problem has the following settings:

a) The service area is the urban area of City of Edmonton, Alberta with a given network previously described in Chapter 4;

b) The O-D travel time is estimated by three different methods: distance based method, ANN model and the A* algorithm. The formal two methods are discussed in Chapter 7 and the A* algorithm is described in Chapter 6;

c) Each vehicle is initially located in the Downtown of the City of Edmonton with a seat capacity of 8 seats/vehicle. Each vehicle's available service time is from 6:00AM to 6:00PM;

d) Customers' trip ends (locations) are randomly generated and uniformly dispersed in the service area. Each customer either has a desired pick-up time or a desired drop-off time (50 percent each in this study) with a required service time between 8:00AM and 12:00AM. The maximum time deviation from each customer's desired time is set to 30 minutes and the maximum ride time is set to 90 minutes.

8.3.2 System Performance Vs. the Objective Function Parameter a_2

The purpose of this section is to examine how the objective function parameter a_2, which is associated with a pick-up or drop-off time deviation, influences the routing and scheduling results under different variations of the O-D travel times in the network. The values of the other modeling parameters are set to $a_1 = 1.0$, $a_3 = 0$, $\alpha = \beta = 90\%$. Figure 8-3 and Figure 8-4 show the relationship between the number of vehicles required and the expected vehicle productivity as functions of the value of parameter a_2 under different values of the coefficient of variation (COV) of the O-D travel time. As would be expected, as the value of a_2 increases, the customers' satisfaction from the service time is weighted higher in the objective function and as a result, more vehicles are required to serve the same number of trips. In addition, a lower vehicle productivity is achieved . It can also be found that in the case of a higher variation of the O-D time, more vehicles are required in order to provide the ame level of service. For example, when the COV increases from zero to 0.10, approximately five extra vehicles are required for the same number of customer trips.

As shown in Figure 8-5, there is an expected strong negative correlation between the average time deviation and a_2. When the a_2 value increases from 0.000 to 0.001, the average time deviation is reduced by approximately 50 percent. It can also be seen that the reduction is lower when the COV of O-D travel time is higher. This may be explained in that under higher variance of O-D travel time, the actual time window at each stop would be narrower and therefore less adjustment can be made to move the schedule closer to the most desired time.

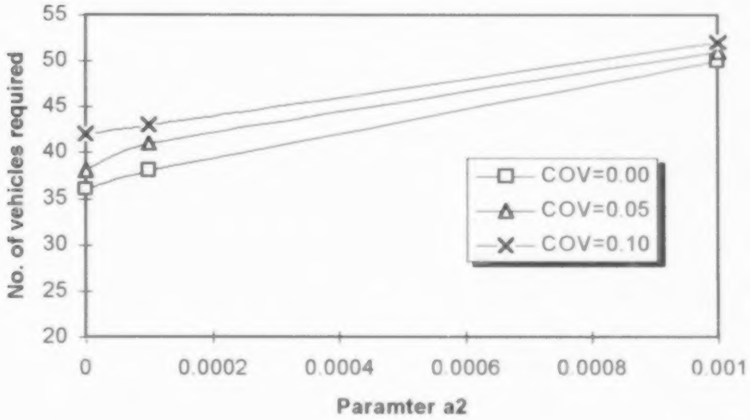


Figure 8-3 The relationship between the number of vehicles required and the parameter a_2

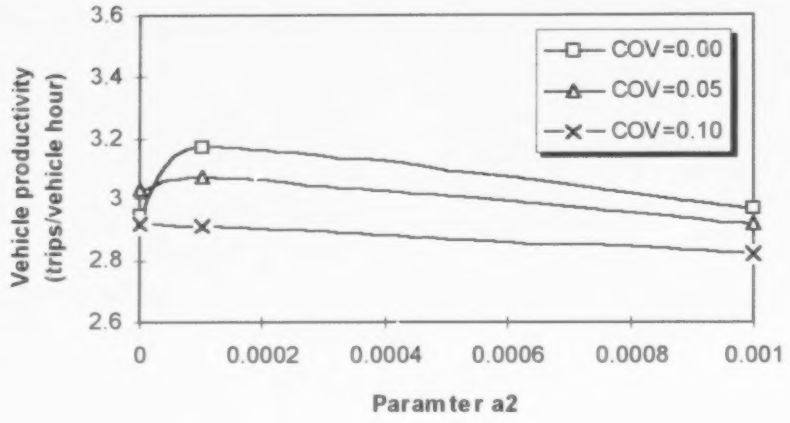


Figure 8-4 The relationship between the expected vehicle productivity and the parameter a_2

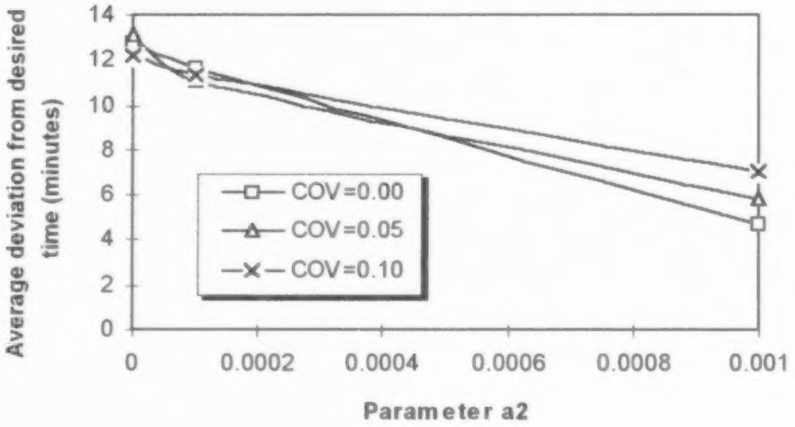


Figure 8-5 The relationship between the expected average time deviation and the parameter a_2

8.3.3 System Performance Vs. Objective Function Parameter a_3

This section examines how the objective function parameter a_3 associated with a customer's excess ride time influences the routing and scheduling results under different variations of the O-D travel times in the network. The values of the other modeling parameters are set to $a_1 = 1.0$, $a_2 = 0$, $\alpha = \beta = 90\%$. Figure 8-6 and Figure 8-7 show the relationship between the number of vehicles required and the vehicle productivity as functions of the value of parameter a_3 under different values of COV of the O-D travel time. It can be found that as the value of a_3 increases, and the customers' satisfaction from the ride time are given a higher weight in the objective function and as a result more vehicles are required to serve the same number of trips and at same time lower vehicle productivity results. It can also be seen from Figure 8-6 that the number of vehicles required is highly sensitive to the a_3 value when it is less than 0.0001. For example, there is an approximately 20 percent increase in the required number of vehicles when the a_3 value increases from 0 to 0.0001. At the same time, the average excess ride time is reduced as much as 50%, as shown in Figure 8-8.

It can also be found from Figure 8-6 and 8-7 that as the O-D time variation (COV) becomes larger, more vehicles are required which results in lower vehicle productivity. For example, when a_3 is equal to 0.0001 and the O-D travel time is deterministic (COV=0), a total of 45 vehicles are required to service the 200 trips with results in vehicle productivity of 3.1 trips/vehicle/hour. However, if the a_3 value is kept the same and the COV of the O-D time is increased to 0.1, 5 extra vehicles are required and the vehicle productivity decreases to 2.95 trips/vehicle/hour.

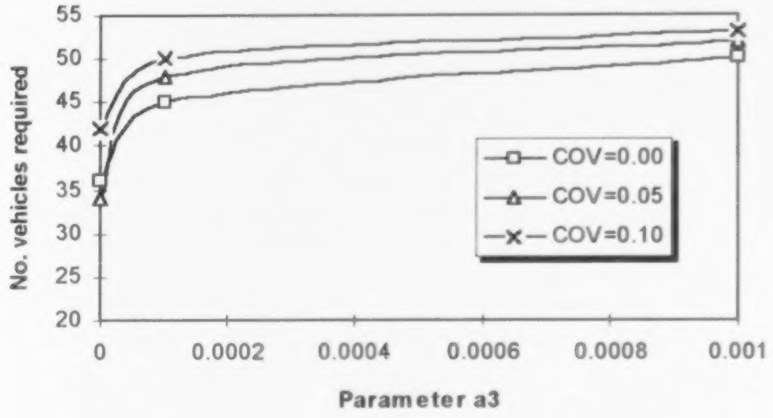


Figure 8-6 The relationship between the number of vehicles required and the parameter a_3

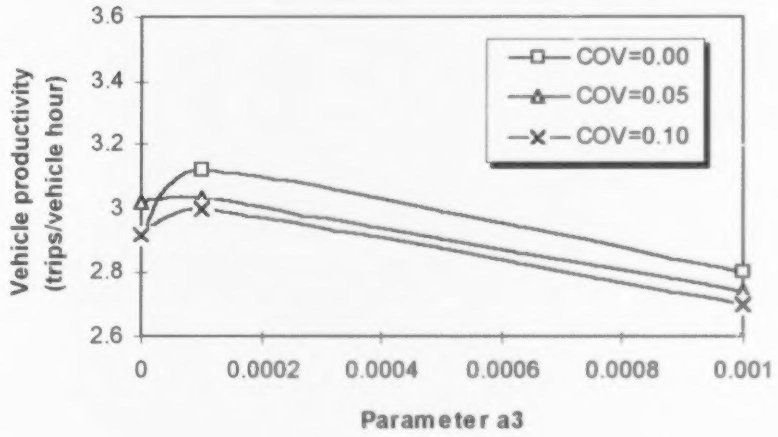


Figure 8-7 The relationship between the expected vehicle productivity and the parameter a_3

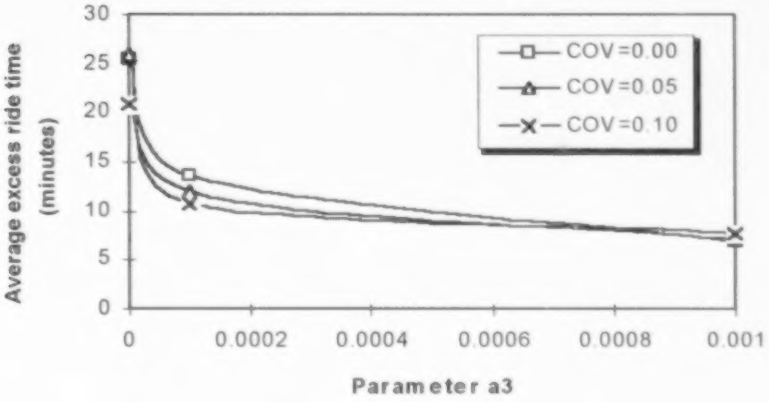


Figure 8-8 The relationship between the expected average excess ride time and the parameter a_3

There is a somewhat changing correlation between the average excess ride time and the O-D time variation. When the a_3 value is less than 0.0008, the smaller value of the average excess ride time corresponds to the larger value of the O-D time COV. When the a_3 value is greater than 0.0008, their correlation becomes less significant.

8.3.4 Computational Efficiency Vs. O-D Travel Time Estimation Methods

This section investigates the computational effort of the proposed algorithm with respect to the O-D travel time estimation methods described in Section 8.2.3. Three different sized dial-a-ride vehicle routing and scheduling problems are used for the analysis. The first problem includes 10 vehicles and 50 customer trips while the second problem involves 40 vehicles and 200 trips. The last problem includes 60 vehicles and

1000 trips. The computational platform is a Pentium 90MHZ. Two routing and scheduling situations discussed in Section 8.1 are considered, that is, the subscriber DARP and real-time DARP. The real-time scheduling situation is simulated by inserting a new trip into the existing schedules after the advanced request DARP of above three problems are solved.

Figure 8-9 shows the relationship between the CPU time required to schedule all the trips and the problem size under different O-D travel time estimation methods. It can be seen that the heuristic shortest path algorithm(A*) is only feasible to be used in the vehicle routing and scheduling process for solving small size problems (less than 50 trips in this test), although it can provide amore accurate estimation of the O-D travel time as compared to other two methods. The routing and scheduling algorithm with the ANN method is much more efficient than one using the heuristic shortest path algorithm. This study shows that the 1000 trips problem is solved within 90 minutes of CPU time when the ANN method is used. This computational time may be still acceptable because for the subscriber trip scheduling, computing time is not as critical as real-time scheduling. As would be expected, using the distance based method is most efficient, however, it is much less accurate than the ANN method , as described in Chapter 7.

Figure 8-10 shows the relationship between the CPU time required to insert a new trip into the existing schedules and the problem size of the existing schedule under different O-D travel time estimation methods. Similar to above conclusion, the heuristic shortest path algorithm (A*) is limited to solving only small size problems. It can be seen that in the case where ANN method is used, it requires less than 10 seconds for the

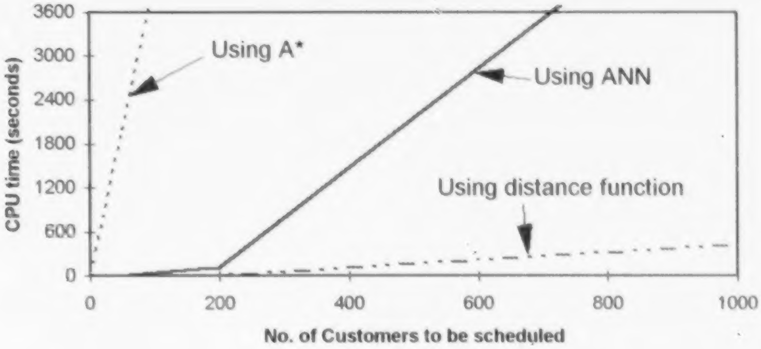


Figure 8-9 Relationship between the computational time and problem size: *subscriber* DARP

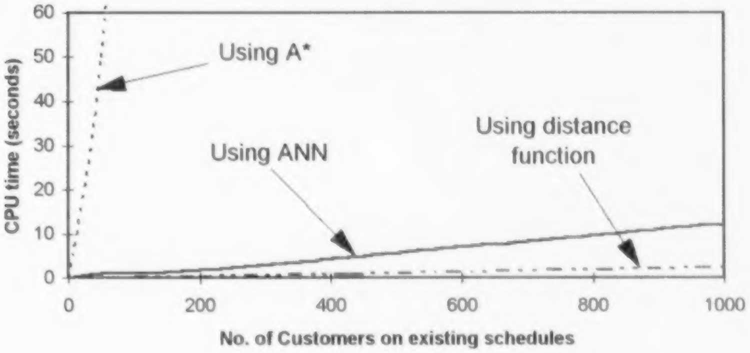


Figure 8-10 Relationship between the computational time and problem size: *real-time* DARP

scheduling algorithm to insert a trip into the existing schedules with 1000 trips on them. This finding implies that the ANN method is efficient enough to be used in a real-time scheduling situation.

8.4 CONCLUSIONS

This chapter discussed how to model the system operator's and customers' objectives in the dial-a-ride vehicle routing and scheduling process when the O-D travel times are modeled as dynamic and stochastic variables. A mathematical formulation of the new DARP is provided. A heuristic dial-a-ride vehicle routing and scheduling procedure is then introduced to solve the new problem. Lastly, a computational study is conducted to illustrate the solution quality and computational requirements of the proposed model and algorithm under different O-D travel time attributes and estimation methods developed in previous chapters. The major conclusions are summarized as follows:

- The O-D travel time variation has significant impact on the vehicle routing and scheduling results. Additional vehicles are required when the influence of the stochastic attribute of the O-D time on the service quality is explicitly considered;
- The impacts of the stochastic O-D travel time on the routing and scheduling results also depends on the modeling parameters which reflect the sensitivity of customers' satisfaction from the service time. For example, the higher the value of parameter a_2 (associated with time deviation from desired time), the higher the impact that the O-D time variation has on the routing and scheduling results;

- Although the explicit consideration of the stochastic attributes of the O-D travel time in routing and scheduling the dial-a-ride vehicle requires extra information on the O-D travel time (for example, mean and variance), it would result in more reliable schedules. The computational performance of the proposed algorithms is acceptable for real applications;
- The computational study shows that the ANN models developed in Chapter 7 are feasible to be used in the dial-a-ride vehicle routing and scheduling algorithm to solve realistic subscriber DARP and real-time DARP (over 1000 trips). On the other hand, the heuristic shortest path algorithm is found to be only viable in solving small sized problems (less than 50 trips);

REFERENCE:

Bodin, L., B. Golden, A. Assad and M. Ball (1983), "Routing and Scheduling of Vehicles and Crews. The State of Art". *Computers and Operations Research* **10** , 69~211.

Jaw, J. Odoni, A. R. , Psaraftis, H. N. and N. H. M. Wilson, (1986), "A Heuristic Algorithm For The Multi-vehicle Advance Request Dial-A-Ride Problem With Time Windows", *Transportation. Research.* -B , Vol. 20B, No. 3, pp. 243-257.

Karian, A. Z. and E. J. Dudewicz, *Modern Statistical, Systems, and GPSS Simulation*. Computer Science Press, New York, 1991.

Psaraftis, H. (1983), "A Dynamic Programming Solution to the Single Vehicle Many-to-many Immediate Request Dial-A-Ride Problem" *Transportation Science* 2, 130~154

Savelsbergh, W. P. M. and M. Sol, (1995), "The General Pickup and Delivery Problem" Transportation Science, Vol. 29, No. 1, 17~29.

Sexton, T. and L. Bodin (1985a), "Optimizing single vehicle many-to-many operation with desired delivery times: I. Scheduling", *Transportation Science* **19**, 378~410.

Sexton, T. and L. Bodin (1985b), "Optimizing single vehicle many-to-many operation with desired delivery times: I. Routing", *Transportation Science* **19**, 378~410.

Wilson, N. H. M. and Weissberg H. *Advanced dial-a-ride algorithms research project: Final report*, Report R76-20, Dept. of Civ. Eng. M.I.T., Cambridge, MA, 1976.

Keeney, R. L. and H. Raiffa, *Decisions with Multiple Objectives*, John Wiley, New York. 1976.

CHAPTER 9

CONCLUSIONS AND RECOMMENDATIONS

9.0 INTRODUCTION

This thesis has systematically investigated two real-time vehicle routing problems, the shortest path problem (SPP) and the dial-a-ride problem (DARP) which arise in two ITS applications: the in-vehicle route guidance systems (RGS) and the automated vehicle dispatching systems (AVDS). New and enhanced vehicle routing problems have been formulated to model the vehicle route optimization problem under dynamic and stochastic link travel time and O-D travel time. Several solution algorithms to these new problems have been developed and used to analyze the influence of the uncertainty of the travel time on the vehicle routing results. This chapter includes two sections. The first section will describe the major conclusions and findings of this research and the second section will discuss recommended research directions.

9.1 CONCLUSIONS

The major conclusions and findings obtained through this research can be summarized in six groups. The first section lists the conclusions arising from the analyses of the dynamic and stochastic link travel times in urban traffic networks. The second section reviews the models developed and conclusions obtained about the estimation of the route travel time in dynamic and stochastic networks. In the third section, findings and conclusions on the dynamic and stochastic shortest path problem (DSSPP) are reviewed. The fourth section summarizes the heuristic shortest path algorithms developed in this thesis and their relative performances in terms of both solution quality and computational efficiency. In the fifth section, the conclusions on dynamic and stochastic O-D travel time estimation using artificial neural networks (ANN) are reviewed. The final section presents the conclusions about the dial-a-ride problem with dynamic and stochastic O-D travel time.

9.1.1 Dynamic and Stochastic Link Travel Time

[1]. This research has shown that the link running time on a link under an uninterrupted, unsaturated flow situation, or approximately has the same type of distribution as the link running speed. Consequently, the distribution parameters of the link running time can be indirectly obtained using data on the link running speed. This finding could be important because traditionally the link running speed is easier to obtain. It is also found, both theoretically and empirically, that the link running time can be represented by a normal or lognormal distribution;

[2]. A simulation model has been developed to analyze the distribution pattern of the delay that a vehicle may experience at a pre-timed signal intersection

under undersaturated traffic conditions. It is found that the delay on a signalized approach is a mixed random variable and its distribution pattern can not be approximated by a single mathematical distribution or distribution family. It is also confirmed that the delay distribution may be bimodal, especially under the situation of poor signal coordination.

The variance of the vehicle delay has been found to be not sensitive to the traffic volume and the quality of progression but to the signal setting. This finding indicates that it may not be necessary to update the vehicle delay variance in real time for an intersection that has a fixed signal timing.

It is also found that when a signalized approach is close to saturated condition, the distribution of the vehicle delay approaches that of a normal distribution.

[3]. A stochastic incident delay estimation model has been developed which explicitly considers the stochastic attribute of the incident duration. The derived formula for the estimation of the mean and variance of the incident delay has been found to be operational in the sense that little extra data and computation effort are required and therefore may be used by a TIC to estimate link travel times in the case of incident congestion. The new model also allows the use of updated information on the incident situation; A computational study showed that the traditional deterministic model may over-estimate or under-estimate the expected incident delay. The estimation error is found to be proportional to the standard deviation of the incident duration. The incident delay has been shown to have a high degree

of variability, even when the expected delay is low. The maximum variance occurs much later than the time when the maximum expected delay occurs.

9.1.2 Route Travel Time in Dynamic and Stochastic Networks

[1]. This research has developed several approximation models to estimate the mean and variance of the route travel time in traffic networks where the link travel times are dynamic and stochastic. It is found that in a dynamic and stochastic network the expected travel time of a given route may not be a summation of the expected link travel times. It also depends on how the average link travel time changes during the day (for example, the second order derivative of the expected link travel time) and link travel time variations;

[2]. The expected travel time of a given route can be accurately estimated using the second order approximation model. Based on the simulation study, the relative improvement of the second order approximation models over the first order model and naive model is quite small (less than 0.2%). However, this small difference could be important for route selection in a traffic network. The relative improvement of the second order model compared to the first order model or naive model depends on the link travel time variance and the severity of the time variation of the link travel time. The larger the link travel time variances and the rate of link travel time change, the greater the relative improvement;

[3]. It has been found that the link travel time distribution pattern has no significant effect on the estimation of the expected route travel time. Therefore, normal distribution may be used to represent the link travel time distribution;

[4]. It has been found that in a dynamic and stochastic network the route travel time variance is not the summation of the link travel time variances as in a traditional naive model. The variance of the route travel times can be more accurately estimated by using the first or second order approximation model. Based on the simulation study, the first order model and the second order model are quite close in terms of solution quality, however, they provide significantly better solutions than the naive model. The relative improvement of the first or second order model compared to the naive model depends on the link travel time variance and the severity of the time variation of the link travel time. The larger the link travel time variances and the link travel time changing rate, the greater the relative improvement.

9.1.3 Shortest Path Problem in Dynamic and Stochastic Networks

[1]. It is theoretically shown that the shortest path problem in dynamic and stochastic networks is computationally intractable and that it cannot be solved exactly using standard shortest path algorithms. This thesis has proposed a heuristic algorithm for solving the DSSPP where the dynamic and stochastic attributes of the link travel times are modeled by the mean and variance of the link travel time as a function of time of day;

[2]. The standard shortest path algorithms that are used to solve deterministic shortest path problems cannot find the minimum expected paths in a dynamic and stochastic network. The solution error by a standard shortest path algorithm in the example used is small primarily because of the simplicity of the network and, more importantly, because the dynamic travel times changed relatively slowly with time.

[3]. The proposed heuristic algorithm improves the solution with only a moderate increase in the overall computation time as compared to a standard shortest path algorithm. The solution improvement is found to be significant when the number of paths (K value) increases from 1 to 2 (more than 18% in terms of the percentage of finding the best solution although the increase in computation time is on the order of 90%);

[4]. As an approximation, the use of standard shortest path algorithms in dynamic and stochastic traffic networks may be acceptable from a practical perspective. This will be especially true if the change of travel time in the network is moderate as in undersaturated networks.

9.1.4 Heuristic Shortest Path Algorithms

[1]. This thesis has developed, implemented and tested several heuristic search methods from the AI field to find the shortest paths in urban traffic networks.

[2]. The branch pruning method proposed in this research uses the information on the estimated maximum travel time from the origin node to destination node and the estimated lower bound of travel time between any pair of nodes to bound the solution search area. It is found that this method can be readily incorporated into the shortest path finding procedure. The resulting heuristic algorithms include parameters and therefore are easily customized to meet the requirement of both computational efficiency and solution quality;

[3]. The traditional A* algorithm can be modified to take into account of the travel speed effect on the travel time estimation (evaluation function) when it is applied to find the shortest path in a traffic network. As a parameterized algorithm, A* algorithms can be adjusted to reach a trade-off between computational efficiency and solution quality;

[4]. In order to overcome the inefficiency of the traditional bi-directional searching algorithm, a modified bi-directional searching method is proposed by introducing a new stopping criteria in the searching procedure. The new algorithm are proven to be very effective to finding the shortest path in a traffic network;

[5]. Among all the heuristic algorithms, the branch pruning label correcting algorithm (BP_LC) generally gave the best results in terms of both computation efficiency and solution quality. Although not guaranteed to

find the optimal routes, the relative error in the route travel time is relative small (0.1%) while its computation saving is significant (30~60%). Although the algorithm B_A*m is slightly slower than the BP_LC algorithm and has the shortcoming of higher estimation error (0.25%), it has an important advantage of being less sensitive to the route time for long trips. This attribute could make it more favorable than the BP_LC in the case where the underlying network is much larger (compared to the Edmonton network).

With an appropriate set of parameter values, the A* algorithm can be guaranteed to provide optimal solutions and provide a beneficial computation time (10%~30% CPU time saving compared to LS). This is in contrast to that of all other heuristic algorithms;

In the situation where the link travel times in the network are dynamic and thus the bi-directional algorithms can't be directly used, the BP_LC and A* algorithm would be the best selections;

9.1.5 Dynamic and Stochastic O-D Travel Time Estimation Using Artificial Neural Networks

[1]. This these has developed several ANN models for estimating the dynamic and stochastic O-D travel time in an urban traffic network. It is found that an ANN model can be trained to effectively map the highly non-linear relationship between the O-D travel time and the location information of the origin and destination nodes in dynamic and stochastic traffic networks;

[2]. The success of an ANN technique for travel time estimation mainly depends on how the input information is abstracted and what type of network model is used. This thesis demonstrated that some enhanced data (for example, distance information) can be very helpful in improving the performance of an ANN. It is found that modeling the mean and variance of the O-D travel time using separate ANN models are more effective than using a joined network ANN model;

[3]. The solution quality of the ANN method is found to be significantly dominant over the traditional regression model for estimating O-D travel times. It can therefore be expected that more reliable solutions can be obtained in the applications by using ANN O-D travel time estimation method instead of the traditional method;

[4]. While the ANN is not as accurate as the shortest path algorithms, it is much faster than the latter. It is empirically shown that the ANN is more than 500 times faster than the shortest path algorithms such as A* and label setting algorithm. Therefore, it is useful in situations where travel time calculations are necessary, but where the computation time is limited. It can be expected that the ANN model holds great potential for various real-time on-line applications such as AVDS.

9.1.6 Dial-A-Ride Vehicle Routing and Scheduling with Dynamic and Stochastic O-D Travel Time

[1]. This thesis has proposed a new dial-a-ride vehicle routing and scheduling model which explicitly incorporates both the system operator's and customers' risk attitudes under dynamic and stochastic O-D travel times. A heuristic routing and scheduling procedure is introduced to solve this new problem.

[2]. It is found that the O-D travel time variation has a significant impact on the vehicle routing and scheduling results. Additional vehicles are required when the stochastic attribute of the O-D time is explicitly considered as compared to a deterministic model. A larger variance in the O-D travel time results in more vehicles required;

[3]. It is also found that the impacts of the stochastic O-D travel time on the routing and scheduling results depend on the model parameters which reflect the sensitivity of customers' satisfaction from the service time. For example, the higher the value of parameter a_2 (associated with time deviation from desired time), the higher the impact that the O-D time variation has on the routing and scheduling results;

[4]. Although the explicit consideration of the stochastic attributes of the O-D travel time in routing and scheduling the dial-a-ride vehicle requires extra information on the O-D travel time (for example, mean and variance), it results in more reliable schedules and thus a better service quality. The

computational performance of the proposed algorithms is found to be acceptable for real applications.

[5]. The computational study shows that the ANN models developed in this thesis are feasible to be used in the dial-a-ride vehicle routing and scheduling algorithm to solve realistic subscriber DARP and real-time DARP (over 1000 trips). On the other side, the heuristic shortest path algorithm is found to be only viable to be used in the dial-a-ride vehicle routing and scheduling algorithm for solving small size problem (less than 50 trips).

9.2 RECOMMENDED FURTHER RESEARCH

This research may be extended in the following five areas.

9.2.1 On Dynamic and Stochastic Link Travel Time

[1]. This thesis has focused on analyzing and modeling the dynamic stochastic travel time patterns on several selected types of links in a urban traffic network. Additional research should be conducted to examine the link travel times under saturated traffic conditions and on other types of links such as those operating under signal coordination. It should be noted that some of the techniques developed in this thesis may be applied to analyze these types of links;

[2]. This thesis has applied various theoretical methodologies in the analysis of link travel time distributions. The next step should focus on collecting field data to verify and calibrate the models developed in this thesis;

[3]. For the incident congestion situation, this thesis proposed a model to incorporate the real-time information on the incident status in the incident delay estimation procedure. This model should be enhanced to consider other types of information such as current traffic volume (or diversion), current queuing length and empirical estimations of incident duration (for example, from police or tow driver). These additions would significantly improve the prediction ability of the proposed model;

[4]. Instead of applying probability theory to model the incident delay as in this thesis, another potential model that should be investigated is using fuzzy set theory to model the incident condition and incident duration. The advantage of this type of model is that it would allow the modeling of one specific type of uncertainty that results from ambiguity and linguistic description;

[5]. It has been shown in this thesis that the link travel time could be significantly stochastic. Consequently, this research should be extended to examine the issues that determines the value of real-time link travel time information and the adequate time interval for link travel time aggregation.

9.2.2 On the Shortest Path Problems in Dynamic and Stochastic Networks

[1]. This thesis analyzed the shortest path problem in a dynamic and stochastic network with the assumption that the individual users' routing objective is to find the paths with the expected minimum travel time (or the users' travel utility function is linear with respect to travel time). Further research may be necessary to analyze other types of shortest path problems such as chance constrained shortest path problem or shortest path problem with non-linear utility functions;

[2]. Another extension to the shortest path problem discussed in this thesis would be to consider the departure time as a decision variable. In this situation, the problem would be to find both the optimal path and the optimal departure time for a specific trip.

9.2.3 On the Heuristic Shortest Path Algorithms

[1]. The heuristic shortest path algorithms introduced in this thesis are exclusively parameterized and their performance under various parameter settings should be tested in real RGS experiments;

[2]. The various heuristic search techniques introduced in this thesis should be adapted to develop heuristic k-shortest path algorithms. The latter is often required to solve some specific categories of shortest path problems such as the dynamic and stochastic shortest path problem discussed in this thesis and shortest path problems with multiple objectives.

9.2.4 On Dynamic and Stochastic O-D Travel Time Estimation Using Artificial Neural Networks (ANN)

[1]. This thesis has demonstrated that the ANN technique is a very effective method to model the dynamic and stochastic O-D travel time in an urban traffic network. The ANN technique may also be used to model other vehicle routing and scheduling related parameters;

[2]. Additional research would be required to develop new on-line training methods for the ANN models so that these models can be improved gradually based on the O-D travel times collected during daily operations of the service vehicles.

[3]. In order to further improve the estimation quality, other types of ANN should also be explored.

9.2.5 On Dial-A-Ride Routing and Scheduling with Dynamic and Stochastic O-D Travel Time

[1]. The new DARP and algorithm should be tested and calibrated using actual data from existing dial-a-ride service systems. Guidelines should be developed for selection of suitable values of the parameters in the model. The algorithm may need to be modified to solve larger sized problems with an acceptable computational time;

[2]. The various O-D travel time estimation methods should be combined in the vehicle routing and scheduling process so that larger problems can be solved without a loss in solution quality;

[3]. Other solution techniques such as genetic algorithm and tabu search methods should be explored to improve the solution of the dial-a-ride problems presented in this thesis.

[4]. Finally, the O-D travel time can be modeled as a fuzzy number and new DARP can be formulated and solved based on fuzzy set theory.

APPENDIX A:

COMPUTATIONAL EFFICIENCY OF THE BRANCH PRUNING ALGORITHM: AN EXPLANATORY MODEL

The purpose of this appendix is to show the computational efficiency of the branch pruning algorithm (BP_LS) proposed in section 6.3.1.1 of chapter 6, as compared to a regular label setting algorithm under an idealized network. The computational time of an algorithm is assumed to be proportional to the search area when the algorithm is used to find a shortest path from an origin node to a destination node. Therefore, the ratio of the search area of the branching pruning algorithm as compared to the search area of a label setting algorithm (LS) is used to measure the computational efficiency of the branch pruning algorithm.

(1) Problem and Notation

Assume that there is a uniform infinite grid network with Euclidean distance used as link cost (Figure A-1). The problem is to derive the relative computation time (or search area) to find the shortest distance path from an origin node to a destination node in this network by the branch pruning algorithm and the regular label setting algorithm. The origin node o is set at the origin of the coordinate axis, (0,0). The destination node r is assumed to be at (x_0,y_0). Assume that L(i) is the shortest travel distance from the origin node to the node i. The Euclidean distance is assumed to be used as the lower bound of the travel distance for the branch pruning algorithm. In addition, note the Euclidean distance from node i to node j as e(i,j) and the respective upper bound of the travel distance as E(i,j). The following sections show how the search area by each algorithm is determined and how the computational efficiency is derived.

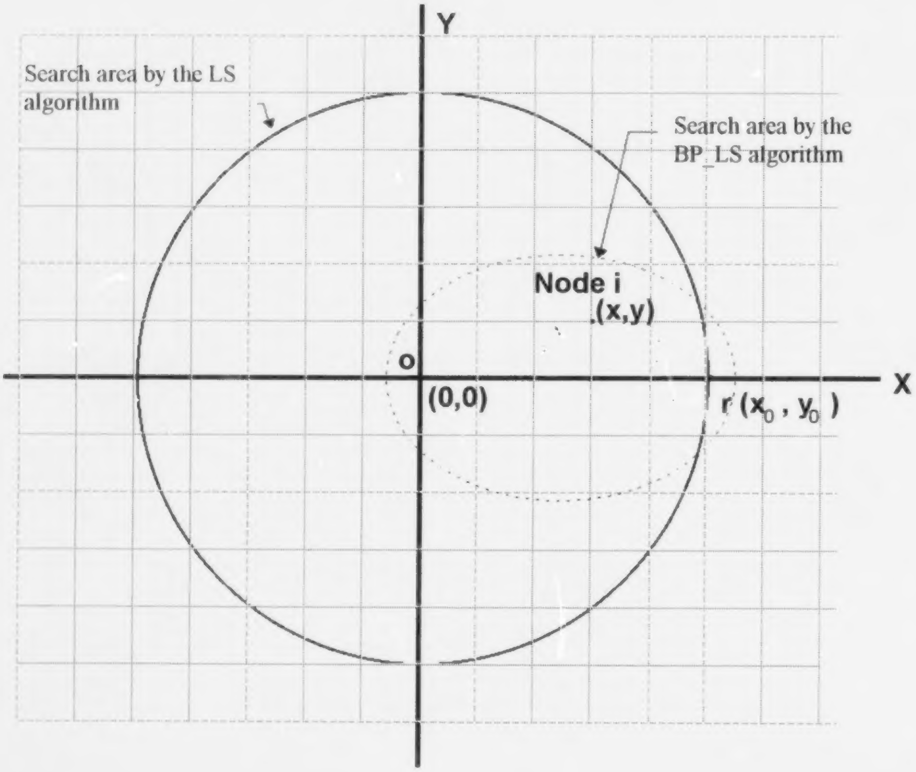

Y
Search area by the LS algorithm
Search area by the BP_LS algorithm
Node i
(x,y)
o
(0,0)
X
r (x0 , y0)

Figure A-1 A schematic illustration of the search area of the LS algorithm and BP_LS algorithm in an idealized network

(2). The Search Area of the LS Algorithm (A_{LS})

In a label setting algorithm, a node i will be set if and only if it is closer to the origin than the destination, that is:

$$iff \qquad L(i) \leq L(r)$$

The area defined by the above inequality is the area that a label setting algorithm must examine before it finds the shortest path. This area can be approximated by the following equation:

$$A_{LS} = \pi \left(x_0^2 + y_0^2\right) \qquad \text{(a-1)}$$

(3) The Search Area of the BP_LS Algorithm (A_{BP})

In the branch pruning algorithm, a node i will be set if and only if the following condition is satisfied:

$$L(i) + e(i,d) \leq E(o,d) \qquad \text{(a-2)}$$

The area defined by the above inequality, i.e., A_{BP}, must be less than the area, noted as A'_{BP}, defined by:

$$e(o,i) + e(i,d) \leq E(o,d) \qquad \text{(a-3)}$$

This inequality function specifies an ellipse defined by the following equation:

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1 \tag{a-4}$$

where a and b are defined as follows

$$2a = E(o,d) \text{ and } 2b = \sqrt{E^2_{(o,d)} - e^2_{(o,d)}}$$

So, $$A'_{BP} = \pi ab = \frac{\pi\ E(o,d)\sqrt{E^2_{(o,d)} - e^2_{(o,d)}}}{4} \tag{a-5}$$

If we define E(o,d) = K e(o,d) where K is the bound factor as described in Section 6.3.1.1 of Chapter 6, then,

$$A'_{BP} = \frac{\pi\ K\sqrt{K^2 - 1}\ e^2_{(o,d)}}{4} = \frac{\pi\ K\sqrt{K^2 - 1}\ (x_0^2 + y_0^2)}{4} \tag{a-6}$$

(4) Computational Efficiency (Ψ)

The computational efficiency of the branch pruning algorithm is defined as the ratio of the search area of the branch pruning algorithm as compared to the search area of a label setting algorithm. Based on this definition, Ψ can be estimated by:

$$\Psi = \frac{A_{BP}}{A_{LS}} \le \frac{A'_{BP}}{A_{LS}} = \frac{K\sqrt{K^2 - 1}(x_0^2 + y_0^2)}{4(x_0^2 + y_0^2)} = \frac{K\sqrt{K^2 - 1}}{4} \tag{a-7}$$

As would be expected, the computational efficiency of the branch pruning algorithm relates to the K value. As shown in Figure A-2, the lower the K value, the larger the computational saving. For example, if the K value is set to 1.5 the resulting computational efficiency is 0.42. That means the branch pruning algorithm only examines

42 percent of the areas that a label setting algorithm does. It should be kept in mind that although using a smaller value of K will increase the computational efficiency of the algorithm, it may reduce the solution quality of the heuristic algorithm as discussed in Chapter 6.

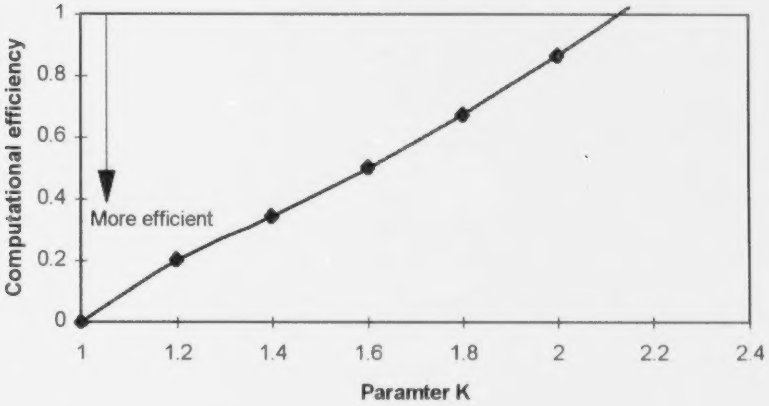


Figure A-2 Computational efficiency of the branch pruning algorithm

APPENDIX B:

COMPUTATIONAL EFFICIENCY OF THE HIERARCHICAL SEARCH ALGORITHM: AN EXPLANATORY MODEL

The purpose of this appendix is to show the computational efficiency of the hierarchical search algorithm discussed in Section 6.3.3.1 of Chapter 6, as compared to a non-hierarchical label setting algorithm for one-to-one shortest path research under an idealized network. The computational time of the algorithms is assumed to be proportional to the number of nodes set when the algorithms are used to find a shortest path from an origin node to a destination node. Therefore, the ratio of the number of nodes examined by the hierarchical search algorithm to the number of nodes examined by a non-hierarchical label setting algorithm (LS) is used in this thesis as a measure of the computational efficiency of the hierarchical search algorithm. The following sections illustrate how this ratio is determined.

(1) Problem and Notation

In order to avoid the effect of the network boundary on the performance of the shortest path algorithm, a uniform infinitely large grid network is used for analysis purposes. The network may be considered to have two type of roads, see freeways and arterials, and can be categorized into two levels, a base level network including both freeways and arterials and an abstract level network including only the freeways, as shown in Figure B-1. Each link in the base level network has a length noted as l, and consequently the network has a node density of $1/l^2$. The abstract level network has an interval distance noted as L, and thus has a network node density equal to $1/L^2$. The problem is to derive the relative computation time (or number of nodes examined) to find the shortest path from an origin node to destination node in the network by the

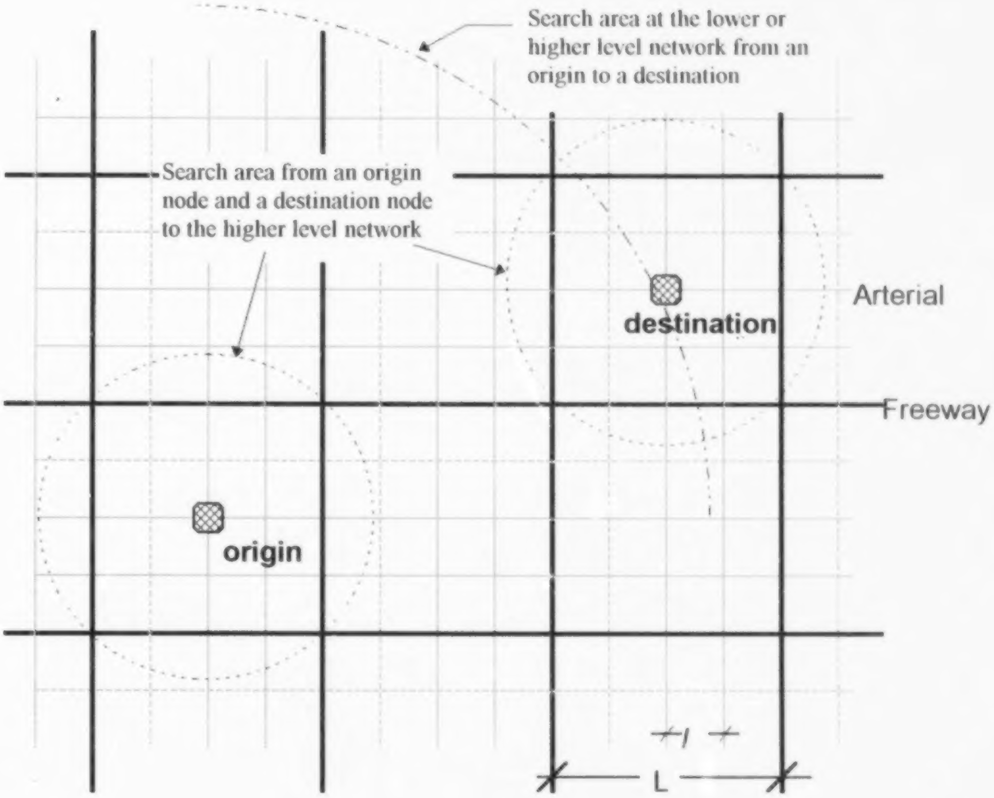

Search area at the lower or higher level network from an origin to a destination
Search area from an origin node and a destination node to the higher level network
destination
Arterial
Freeway
origin
l
L

Figure B-1. A hierarchical Euclidean network hierarchical search algorithm and by a non-hierarchical label setting algorithm. The origin node and the destination node are assumed to be located inside different grids of the abstract level network. The Euclidean distance from the origin node to the destination node is noted as R.

The number of nodes examined by each algorithm can be determined based on the search area examined by the algorithm and the network node density of the underlying network with the following relationship:

Estimate of # of nodes examined ≈ Search area * average density of the search area

The following sections is the derivation procedure of the search area by each algorithm.

(2) The Number of Nodes Examined Using a Non-hierarchical Label Setting Algorithm (N_0)

The search area of the non-hierarchical label setting algorithm is defined by the circle centered at the origin with radius equal to the distance between the origin and destination (R), as shown in Figure B-1. For the non-hierarchical search procedure, the search is undertaken on the base level network with a density of $1/l^2$. Therefore, N_0 can be obtained by:

$$N_0 = \pi \cdot R^2 \cdot \frac{1}{l^2} \tag{b-1}$$

(3) The Number of Nodes Examined Using a Hierarchical Label Setting Algorithm (N_H)

As discussed in Section 6.3.3.1 of Chapter 6. the hierarchical search is composed of two procedures. The first procedure is to examine the area around the origin and destination node on the base level network (the small circles shown in Figure B-1). The second procedure is to examine the area from the origin node to the destination node on the abstract level network (the big circle in Figure B-1).

The number of nodes examined in the first procedure (N_{od}) can be decided by:

$$N_{od} = 2 \cdot (\frac{L}{\sqrt{2}})^2 \cdot \pi \cdot \frac{1}{l^2} = \frac{\pi L^2}{l^2} \tag{b-2}$$

The number of nodes examined in the second procedure (N_a):

$$N_a = \pi \cdot R^2 \cdot \frac{1}{L^2} \tag{b-3}$$

The total number of nodes can be obtained by (N_H):

$$N_H = N_{od} + N_a \tag{b-4}$$

(4) Computational Efficiency (ψ)

Based equations (b-1),(b-2),(b-3) and (b-4), the computational efficiency of the hierarchical searching method can be calculated by:

$$\psi = \frac{N_H}{N_0} = \frac{\pi \cdot \frac{L^2}{l^2} + \pi \cdot R^2 \cdot \frac{1}{L^2}}{\pi \cdot R^2 \cdot \frac{1}{l^2}}$$

or

$$\psi = \frac{L^2}{R^2} + \frac{l^2}{L^2} \tag{b-5}$$

As expected, Equation (b-5) shows that the further away the origin node is from the destination node (or smaller the R/L value is) and the higher the network is abstracted (or the higher the L/*l* is), the more effective the hierarchical searching algorithm will be. Figure B-2 shows the numerical relationship between the computational efficiency and the relative trip length as compared to the interval distance under different network density ratio (*l/L*). It can be found that when the trip length (R) covers more than two freeway intervals (or R/L>2), the increase of trips length only generates very slight computational saving. It should be noted that the upper bound of the computation efficiency is l^2/L^2 (that is, when r approaches to infinite).

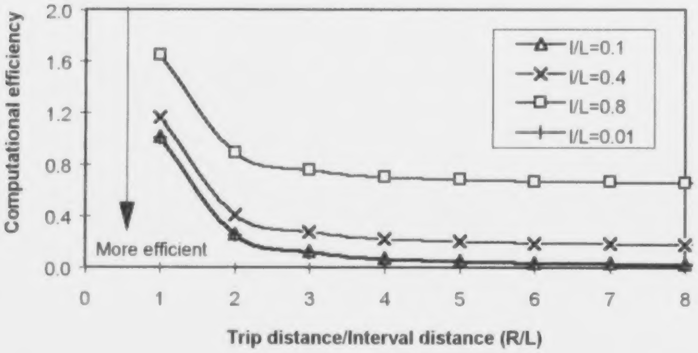


Figure B-2. The computational efficiency of the hierarchical label setting algorithm

(5) Numerical Example

Assume: $l = 100$ (m); $R = 10{,}000$ (m)

$L = 1{,}000$ (m)

Based on Equation (b-5), the computational efficiency (ψ) is 0.02. That means the hierarchical searching method only examines two percent of the nodes that a non-hierarchical searching algorithm examines.

APPENDIX C:

ARTIFICIAL NEURAL NETWORK(ANN): AN INTRODUCTION

An ANN is fundamentally a information processor and it can been trained to perform a variety of tasks. They have been used successfully in image processing, speech recognition and solving combinatorial problems. The popularity and successes of ANN technologies are directly attributed to theirs architecture of easily mapping to a parallel computation model, its ability to infer patterns from data that is incomplete and/or inaccurate, and its special characteristics in knowledge representation and acquisition.

An ANN consists of a number of simple process elements (PE) linked together via weighted directed connections as shown in Figure C-1. Each PE (i) receives an input signal, Xj, from the other PE (j) via incoming connections. There is a weight associated with each connection, denoted by w_{ij}, which indicates the "effect" of PE (j) on PE (I). The input signals are combined together and yield a *net value* (T_i) by a basic function such as the following commonly used linear function:

$$T_i = \sum_j w_{ij} X_j \qquad \text{(c-1)}$$

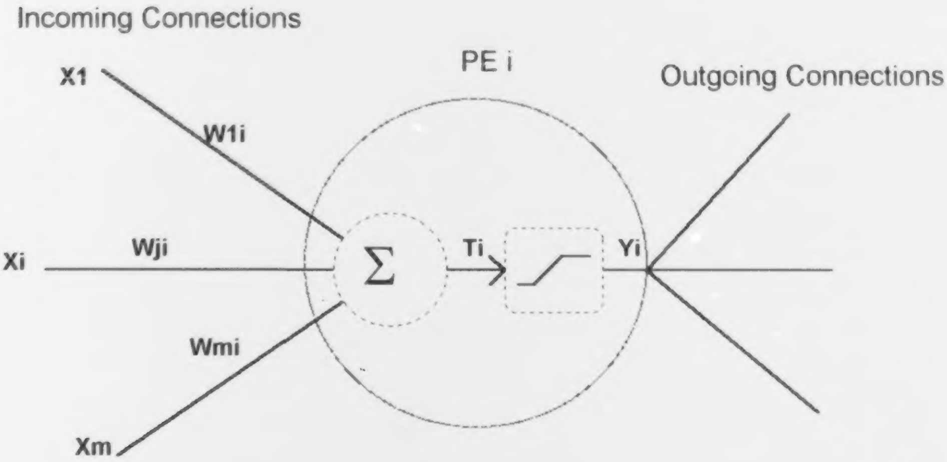

Incoming Connections
PE i
Outgoing Connections
X1
W1i
Xi
Wji
Σ
Ti
Yi
Wmi
Xm

Figure C-1 A typical ANN processing element (PE)

This net value is then transformed into an *activation value,* Yi, as an output of the PE i by a nonlinear *activation function* such as the sigmoid function:

$$Y_i = \frac{1}{1+e^{-T_i^2}} \qquad \text{(c-2)}$$

The output value is then sent to all the PEs which has outgoing connections associated with it.

An ANN can simply be viewed as a model with input and output channels. The input value to an ANN propagates through the ANN until yielding an output of the ANN. An ANN can learn by adjusting the connection weights (w_{ij}) based on examples, i.e., pairs of input with desired output. The objective of the learning stage is to find the weights to minimize an error function representing the difference between the actual output with desired output. An error function is commonly defined as a sum of squared errors over all the output cells,

$$E = 0.5 \sum_{k} (Y_k - D_k)^2 \tag{c-3}$$

where: Y_k = the actual output on output node k, D_k is the desired output for output node k.

The process of identifying the optimal weights can be completed by using many algorithms. One of the most popular learning algorithm is the back-propagation method for feed forward neural networks. The back-propagation algorithm is essentially a gradient decent search algorithm in which the weight adjustments are determined by the error signals transmitted in the backward direction. For each training example, the input attributes are fed in from the input layer and the output of each PE can be calculated layer by layer, from the input layer to the output layer, using equations (c-1) and (c-2). The learning error can then be determined by comparing the output of each cell on output layer (Y_k) to the desired output response, D_k. The amount of error attributed to each cell, δ_j, is calculated layer by layer, from output layer to input layer, by using following formula:

$$\delta_j = \begin{cases} (D_j - Y_j) \cdot Y_j \cdot f'(T_j) & \text{for output PEs} \\ \sum_{m > j} w_{jm} \cdot \delta_m \cdot f'(T_j) & \text{for other PEs} \end{cases} \tag{c-4}$$

where: T_j = defined in Equation (c-1);

$f(T_j)$ = Y_i, defined in Equation (c-2);

$f'(T_j)$ = first derivative, = $Y_j\,(1-Y_j)$

After the δ_i associated with each PE is calculated, every weight is adjusted by the following equation:

$$w_{ij}(n) = w_{ij}(n-1) + \eta \cdot \delta_j \cdot Y_j + \alpha \cdot \Delta w_{ij}(n-1) \qquad \text{(c-5)}$$

where: η = learning rate;

α = momentum;

$\Delta w_{ij}(n-1)$ = the weight change at iteration n-1.

The learning rate and the momentum are external parameters which can be adjusted to improve the training effectiveness and efficiency.

GLOSSARY

A* algorithm: A ***heuristic shortest path algorithm*** that examines nodes in order of their "likelihood" of being on the minimum path. The nodes that have a higher "likelihood" of being on the minimum path are given priority over those with a lower "likelihood" during search procedure. The "likelihood" of a node being on the minimum path is defined as the summation of the cost from the origin node to this node plus an estimated cost from this node to the destination node

Bellman's "principle of optimality": A principle that is used as the foundation of dynamic programming theory. It is stated as: an optimal decision has the property that, whatever the initial decision is, the remaining decisions must be optimal with respect to the outcome resulting from the first decision. If applied in shortest path search, it can be simply interpreted as, any segment of a shortest path is a shortest path from the beginning node of that segment to the ending node of that segment

Bi-directional search algorithm: A ***heuristic shortest path algorithm*** that is composed of two simultaneous search procedures. One search procedure proceeds forward from the origin node while another search procedure proceeds backward from the destination node. The shortest paths are identified when these two search procedures meet at some middle node(s)

Branch pruning algorithm: A ***heuristic shortest path algorithm*** that limits the search area by pruning the intermediate nodes that have a lower likelihood of being on the shortest paths to the destination node

Centralized RGS: A type of RGS architecture where the RGS route for an individual vehicle is calculated externally to the vehicle in a central location (usually the ***TIC***)

Coefficient of variation (COV): The standard deviation of a random variable divided by the mean of the random variable

DARP: Dial-a-ride problem, a problem arising in a dial-a-ride system where customers call a dispatcher in order to request service. Each customer specifies a distinct pick-up and drop-off location in the service area and usually a desired time for pick-up or drop-off. The problem is to develop a set of "optimal" routes and schedules for vehicles to carry the customers from their pick-up locations to their drop-off locations. DARP can be classified into two problems: ***subscriber DARP*** and ***real-time DARP***

Deterministic network: A network where all the links have or are assumed to have ***deterministic travel times***

Deterministic travel time: A travel time with a determined quantity

Distance function: A function representing the relationship between the travel distance between two geographical locations and the coordinates of these two locations, for example, Euclidean distance function and Manhattan distance function

Distributed RGS: A type of RGS architecture where the RGS route is calculate by a computer in the vehicle based on information sent from the ***TIC***. The route calculation is made regardless of what any other RGS vehicles are doing

Dynamic and stochastic network: A network where some of the links have or are assumed to have ***dynamic and stochastic travel times***

Dynamic and stochastic travel time: A travel time which is a stochastic process. That is, throughout the day, the travel time is a random variable with a determined probability distribution

Dynamic minimum path: A minimum path that is calculated based on estimates of link travel times that are a function of when the driver is estimated to arrive at a particular link as opposed to current estimates of the links current travel time

Dynamic network: A network where some of the links have or are assumed to have dynamic travel times

Dynamic shortest path: See ***Dynamic minimum path***

Dynamic travel time: A ***deterministic travel time*** which is a function of the time when a vehicle enters the link, or the time of day

Expected minimum path: The path with minimum expected travel time from an origin to a destination with a given departure time in a network

Heuristic shortest path algorithm: A shortest path algorithm that uses ***heuristic*** during its search procedure

Heuristic: A rule of thumb, strategy, trick, simplification, or any other kind of device which drastically limits the search for optimal solutions in large problem spaces. Heuristics do not guarantee optimal solutions; it is considered useful if it offers solutions which are good enough most of the time

K-shortest path algorithm: A solution procedure that is used to find the shortest, the second shortest and up to k*th* shortest paths from an origin node to a destination node in a network

Link: A representation of a section of roadway that connects two nodes and has the same set of characteristics (for example, speed limit, capacity)

Link travel time: See ***travel time***

Node: A representation of a physical location of traffic network (for example, intersections)

NP-hard: A class of problems for which no ***polynomially-bounded algorithm*** has yet been found. It has been suggested (but not proved) that the effort required to solve this class of problems increases exponentially with problem size in worst case. ***Heuristic*** or approximate procedures are commonly resorted to obtain near-optimal solutions to a problem proved to be NP-hard.

O-D travel time: Time spend to travel from an origin location to a destination location in an urban traffic network using a specific transportation mode such as car and transit

Optimal path: See ***Expected minimum path***

Path: Represents a list of sequential links from an origin node to a destination node

Polynomially-bounded algorithm: A procedure whose computational time increases only polynomially with problem size in the worst case. The class of all problems for which polynomially-bounded algorithms are known to exist is denoted by ***P***.

Real-time DARP: One type of DARP. In this problem all the customers demand immediate service, the routing and scheduling is done in real-time

Route: 1) See ***Path***; or 2) A sequence of locations (for example, pickup and/or drop-off locations) to be visited

Schedule: Specifies the times at which the activities at specific locations (e.g. pick up or drop off a customer) are to be carry out

Standard shortest path algorithms: The shortest path algorithms that are used to find the shortest paths in deterministic networks

Stochastic network: A network where some of the links have or are assumed to have ***stochastic travel times***

Stochastic travel time: A travel time that has some random component. In this thesis the travel time is represented by a random variable

Subscriber DARP: One type of DARP. In this problem the customers call in advance, and therefore a complete database of customer demand is known before any routing and scheduling is carried out

Time window: A specific time interval during which a service task (for example, pick-up or drop-off a customer) is required to be completed

TIC: Traffic Information Center, a place where traffic network link travel time information is stored and disseminated to RGS vehicles or AVDS operation center

Utility function: A numerical function representing the relationship between the degree of satisfaction (or dissatisfaction) that an individual or group (for example, customers) associated with a series of decisions or alternatives